Animation Adaptative de Chevelures Virtuelles

Florence Bertails

Troisième année au département Télécommunications de l'INPG
DEA Imagerie, Vision, Robotique

Animation adaptative de chevelures virtuelles

Florence BERTAILS

Projet effectué sous la responsabilité de
Marie-Paule CANI

Dans le cadre du projet iMAGIS (laboratoire GRAVIR/IMAG)

---

juin 2002

Remerciements

Je remercie tout d'abord Marie-Paule Cani pour m'avoir encadrée pour ce projet. L'animation de chevelures en images de synthèse est un sujet qui m'intéresse beaucoup, et je suis heureuse d'avoir pu travailler dans ce domaine sous sa responsabilité.
Mes remerciements vont également à toutes les personnes d'*i*MAGIS qui m'ont apporté leur aide durant ce projet. Je pense en particulier à Gilles, qui a facilité la tâche à tous les stagiaires de DEA en développant la bibliothèque du *Viewer*, mais aussi à François, David, et aux deux Sylvain, pour leurs nombreux conseils.
Enfin, je tiens tout particulièrement à remercier Jean-Marc Dolmazon, directeur du département Télécom de l'INPG, grâce à qui j'ai eu la possibilité de suivre le DEA Imagerie, Vision et Robotique en double cursus cette année. Merci d'avoir compris à quel point je tenais à faire ce DEA, et de m'avoir témoignée autant de confiance.

Abstract

This report presents the internship I have been doing since december 2001 for both my third year of telecommunications engineering school, and my first year of research oriented studies in Robotics, Computer Vision and Image Processing.
This internship took place within iMAGIS research team (part of GRAVIR laboratory, and specialized in computer graphics), on *Adaptative Animation of Virtual Hair*.
The first part of the work consists in overviewing the existing techniques developped for hair animation, and physics based animation. Then, we propose a new hair animation technique based on adaptative subdivisions and fusions of wisps, aiming at giving real-time simulations. Finally, we give results and conclude with future work.

Table des matières

Introduction

L'animation réaliste de chevelures virtuelles est un sujet de recherche et de développement qui suscite actuellement un intérêt très fort, aussi bien de la part de la communauté des chercheurs travaillant dans le domaine de l'image de synthèse, que des entreprises spécialisées dans le jeu vidéo ou les films d'animation.
En effet, à l'heure où la création et l'animation de personnages virtuels sont en plein essor, l'animation de la chevelure des personnages en demeure l'une des principales difficultés : cette animation est très complexe, au vu du grand nombre de cheveux qui composent une chevelure (environ 100 000) ; de plus, pour être réaliste, et permettre de gérer le mouvement de chevelures les plus diverses possibles (courtes, longues, lisses, bouclées etc.), une telle animation doit prendre en compte les interactions multiples qui se produisent lors du mouvement, à la fois au sein-même de la chevelure, et entre le corps et la chevelure.
A titre d'illustration, le film d'animation *Final Fantasy*, sorti dans les salles en été 2001, met en scène des personnages virtuels très réalistes, et, pour justement accroître le réalisme, un maximum d'efforts se sont concentrés sur l'animation de la chevelure de l'héroïne ; cette animation a nécessité environ 80% du temps total d'animation pour ce personnage, alors que sa chevelure est très lisse, et assez courte (cf. fig. 1). Malgré le réalisme évident de cette animation, on est donc encore loin d'un résultat économique en temps de calcul. De plus, l'animation d'une chevelure longue, et moins lisse, serait encore beaucoup plus complexe, en raison des nombreuses interactions et collisions qu'il faudrait prendre en compte.

Une difficulté supplémentaire intervient dans les domaines d'application où l'on on impose, en plus, la contrainte de temps-réel[1] : cela apparaît alors comme un facteur très limitant puisqu'il s'agit de trouver un compromis entre réalisme de l'animation et coût en temps de calcul. Malgré les progrès fulgurants réalisés par le hardware graphique ces dernières années, la puissance des machines ne suffit pas du tout à remplir les objectifs souhaités : actuellement, l'animation de chevelures dans les jeux vidéo est réalisée de manière très simpliste (les personnages ont souvent des cheveux courts, ou bien attachés), afin de ne pas monopoliser le temps de calcul, et ainsi de ne pas compromettre l'interactivité du jeu.

[1] Une animation est dite *en temps r´eel* lorsqu'elle se d´eroule de mani`ere simultan´ee avec les calculs qui permettent de la g´en´erer.

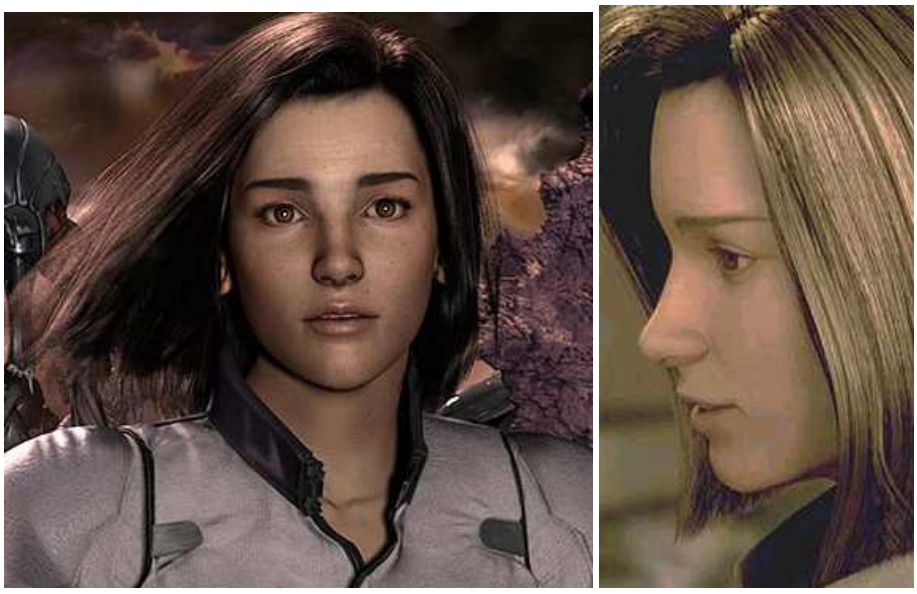


FIG. 1 – Héroïne virtuelle du film *Final Fantasy*

Ainsi, si de nombreux travaux ont porté sur la modélisation du comportement, des mouvements, des déformations corporelles et des vêtements de ces personnages, l'animation de chevelures a longtemps été laissée de côté.
Cependant, des travaux de recherche récents ont abouti à l'animation de chevelures quelconques de manière très réaliste, en proposant de structurer la chevelure en mèches interagissantes. Chacune de ces mèches est représentée par une hiérarchie de modèles : un squelette linéique pour l'intégration de la dynamique du mouvement, une enveloppe souple permettant de prendre en compte les déformations de la mèche, et des cheveux individuels tracés dans cette enveloppe pour le rendu de la chevelure. Des collisions anisotropes permettent alors de prendre en compte l'ensemble des interactions en présence. Si elle permet de simuler de manière réaliste le mouvement d'une longue chevelure, l'approche précédente reste très lourde en temps de calcul (plusieurs heures de calcul pour quelques secondes d'animation).
Dans ce projet, nous développons une nouvelle méthode d'animation de chevelure basée sur une répartition adaptative de la complexité de la chevelure, grâce à un double mécanisme de subdivisions et de fusions des mèches constituant le modèle. Cette méthode permet de concentrer le temps de calcul de chaque image sur les zones en mouvement, où la chevelure est finement subdivisée, tandis que les parties au repos nécessitent moins de calculs. Grâce à cette nouvelle méthode de gestion automatique de la complexité d'une chevelure, on est capable d'effectuer des animations proches du temps réel, sans que le réalisme soit trop dégradé.

La première partie de ce rapport fait la synthèse des techniques antérieures en animation de chevelure, et introduit la technique d'animation par modèles physiques.
Dans la seconde partie, on développe notre méthode d'animation de chevelures, basée sur la subdivision et la fusion adaptatives de mèches.
Enfin, la troisième et dernière partie présente et analyse les résultats obtenus.
En conclusion, on dresse un bilan de la méthode et on évoque les travaux futurs.

Chapitre 1

Etat de l'art

Synthétiser de manière réaliste une chevelure en mouvement suppose de résoudre les trois problèmes suivants : la *modélisation de la coiffure*, ainsi que l'*animation* et le *rendu* de la chevelure[1]. Si des méthodes satisfaisantes de rendu ont été développées avec succès il y a déjà un certain nombre d'années [8], la recherche concernant la modélisation et l'animation de chevelures commence à peine à fournir des résultats intéressants. En effet, étant donné le grand nombre de cheveux qui composent une chevelure (environ 100 000), il a semblé longtemps impossible de parvenir à une simulation à la fois réaliste et peu coûteuse en temps de calcul. Ce n'est que depuis très récemment que des méthodes permettent de simuler de manière réaliste le mouvement d'une chevelure, en tenant précisément compte de l'ensemble des interactions en présence.
Dans cette partie, la première section présente, à travers la critique de références bibliographiques, l'ensemble des travaux réalisés dans la recherche sur l'animation de chevelures en informatique graphique.
La seconde section a pour but de décrire une technique d'animation très adaptée à l'animation de phénomènes naturels complexes : l'animation par *modèles physiques*.

1.1 Travaux antérieurs en animation de chevelures

Introduction

Les premières méthodes pour animer efficacement une chevelure, développées dans les années 90, ne tiennent pas compte de la complexité des interactions en présence, et de manière générale, ne permettent de simuler que des mouvements lents de chevelures à la fois courtes et très lisses.
Quelques travaux ont ensuite proposé d'exploiter la cohésion globale de la chevelure lors du mouvement afin de prendre en compte de manière efficace les interactions entre cheveux ; mais là encore, de telles méthodes ne parviennent pas à restituer les discontinuités importantes que subit une chevelure en fort mouvement, telles que l'on peut les observer sur la figure 1.1.
Enfin, des travaux développés récemment [16], permettent de générer de manière très réaliste le mouvement quelconque d'une chevelure, en tenant précisément compte de l'ensemble des interactions en présence. Cependant on est encore loin de parvenir à simuler de manière réaliste des mouvements de chevelures en temps réel.

[1] Rappelons que dans ce projet, on s'int´eresse essentiellement aux probl`emes li´es `a l'*animation* de chevelures ; toutefois, la mod´elisation, l'animation et le rendu d'une chevelure ne peuvent pas toujours ˆetre facilement d´ecorr´el´es.

FIG. 1.1 – Image réelle d'une chevelure en fort mouvement

1.1.1 Représentation géométrique d'une chevelure

Représentation linéique

Bon nombre de chercheurs [2, 13, 18] ont utilisé une représentation linéique de la chevelure, en modélisant chaque cheveu ou chaque mèche par une chaîne 1D reliant des particules ou des segments 1D. Mais une chevelure qui est uniquement représentée par un ensemble de chaînes linéiques manque d'épaisseur et de volume. Hadap et Magnenat-Thalmann [7] ont utilisé une représentation de chevelures mixte, en modélisant les grandes échelles à l'aide d'une représentation volumique, et les détails plus fins (cheveux individuels) par des structures linéiques.

Représentations surfaciques et volumiques

Koh et Huang [11, 12] et Kim et Neumann [10] ont choisi de représenter la chevelure par un ensemble de bandes 2D de cheveux, chacune de ces bandes étant modélisée par un morceau de surface paramétrique, de type *NURBS*[2].
L'avantage de cette représentation pour l'animation (cf. 1.1.2) est qu'elle permet d'obtenir des déformations très variées de la chevelure, à un coût moindre. Le problème est que, là encore, les chevelures représentées de la sorte manquent de volume. Pour pallier à cet inconvénient, Kim et Neumann [10] ont créé artificiellement du volume aux coiffures qu'ils représentent, en ajoutant un *offset* aux surfaces de la chevelure. Cet ajout de volume permet également de prendre en considération, de manière simplifiée, les interactions entre cheveux (cf. 1.1.3). La méthode [16] propose une représentation volumique plus élaborée de la chevelure, en la modélisant par un ensemble de mèches volumiques déformables.

1.1.2 Techniques d'animation applicables à l'animation d'une chevelure

Il existe deux grandes techniques d'animation : l'animation par *modèles descriptifs* et l'animation par *modèles générateurs*.
La première technique consiste à donner aux objets que l'on souhaite animer une fonction explicite de son mouvement en fonction du temps. C'est le cas par exemple de l'animation par *positions clefs*, ou encore de la capture de mouvement. La technique d'animation par positions clefs a été largement employée pour animer des chevelures.
A l'opposé, les modèles générateurs, plus récents, fournissent un ensemble de lois (si ces lois issues

[2]Non Uniform Rational B-Spline

de la physique, on parle alors de *modèles physiques*) auxquelles l'objet est soumis. Cette technique d'animation (cf. 1.2), a l'avantage de permettre la synthèse de mouvements génériques.
Ci-dessous on présente les principales méthodes qui ont été employées pour la dynamique de chevelures : l'animation par positions clefs (modèle descriptif), l'animation de chaînes articulées (modèle pseudo-physique), l'animation de chaînes masses-ressorts (modèle physique), et l'animation de milieu fluide ou continu (modèle physique).

Animation par positions clefs

L'animation par position clefs est la technique d'animation utilisée traditionnellement dans les dessins animés. Elle consiste à stocker l'information pertinente du mouvement à plusieurs instants donnés dans des clefs, et à interpoler le mouvement entre les clefs.
Les méthodes basées sur la représentation de la chevelure à l'aide de surfaces paramétriques [10, 11] sont bien adaptées à cette technique, car la déformation d'un morceau de surface est simplement lié au déplacement des points contrôlant cette surface. L'animateur souhaitant animer une chevelure de cette façon peut donc construire une série de positions clefs $K_{i\in[0..n]}$, en choisissant pour chacune de ces positions K_{i_0} l'emplacement de l'ensemble des points de contrôle $C_{i_0,p\in[0..m]}$ de la chevelure. L'interpolation entre les différentes positions clefs est ensuite gérée par le système de manière très simple, puisque cela revient à utiliser une fonction interpolant dans le temps chaque suite $S_i^{p_0}$ de points de contrôle $C_{i\in[0..n],p_0}$.
L'inconvénient d'une telle approche, est que les résultats qu'elle produit ne sont valables que pour le mouvement spécifiquement représenté. De manière plus générale, l'opposition entre les modèles descriptifs du mouvement et les modèles générateurs du mouvement fera l'objet d'un paragraphe en 1.2.

Animation de chaînes articulées

Un certain nombre de chercheurs [2, 13] ont choisi d'utiliser la méthode initialement proposée par Anjyo *et al.*, qui consiste à représenter chaque cheveu par une *chaîne de segments rigides articulés*. Cette méthode sert à la fois à obtenir la position initiale des cheveux (modélisation de coiffure), et à les animer. Dans le cas de la représentation des cheveux par des chaînes articulées [9], la position et l'orientation de chaque segment peuvent être calculées en cascade, des racines jusqu'aux pointes, en utilisant des équations analytiques unidimensionnelles prenant en compte la gravité. L'inconvénient majeur de cette méthode, comme l'illustre la figure 1.2 est que la réponse à la collision d'un cheveu avec le corps a une incidence uniquement sur le segment en collision (et sur les segments suivants), mais n'est pas propagée au niveau des segments proches de la tête, puisque la modification de position et d'orientation des segments ne se fait que dans un sens (racines vers les pointes). Cette méthode est donc satisfaisante dans le cas où la chevelure considérée est courte, mais elle ne marcherait pas pour une chevelure longue atteignant les épaules : comme le montre la figure 1.2, la réponse à une telle collision ne serait pas du tout naturelle.

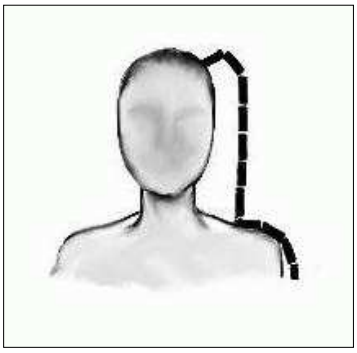


FIG. 1.2 – Collision entre une chaîne articulée
et un membre du corps autre que la tête

Récemment, Lee et Ko [13], ont étendu la méthode et proposé de résoudre le problème en ajoutant artificiellement des forces de synthèse aux segments se trouvant à proximité du segment en collision (cf. fig 1.3). Cette méthode permet alors, de manière artificielle, de tenir compte de la propriété du cheveu à se déformer sur toute sa longueur lors d'une collision avec un objet autre que la tête.

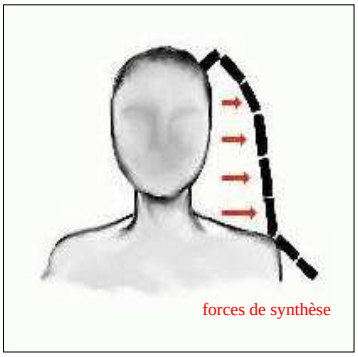


FIG. 1.3 – Collision entre une chaîne articulée et le corps
avec ajout de forces de synthèse correctives

Animation de systèmes masse-ressort

Une manière de modéliser à l'intérieur même du système dynamique la capacité d'un cheveu à se déformer sur toute sa longueur consiste [12, 15, 16] à représenter chaque cheveu par une chaîne de masses-ressorts. La réponse à une quelconque collision entre ce système et le corps est conforme à la réalité (cf. fig. 1.4).
Cependant, un tel système présente deux inconvénients par rapport aux chaînes articulées : d'une part, la longueur d'une chaîne de masses-ressorts est variable, alors qu'un cheveu est inextensible ; d'autre part, l'intégration dans le temps d'un tel système nécessite beaucoup de calculs, car chaque particule subit des forces élastiques de la part de ses deux particules voisines. Rosenblum *et al* [18] ont été parmi les premiers à utiliser des chaînes de masses-ressorts pour animer une chevelure ; dans leur méthode, chaque cheveu est modélisé par une chaîne comportant une quinzaine de masses reliées entre elles par des ressorts amortis. Mais avec une telle méthode, il n'est pas possible d'animer plus

d'un millier de cheveu sans générer des calculs trop lourds.
Les systèmes masses-ressorts sont donc plutôt adaptés à l'animation d'une chevelure divisée en mèches : chaque mèche de cheveu est en effet relativement extensible, et le nombre de mèches composant une chevelure peut-être choisi suffisamment petit pour ne pas générer des calculs trop coûteux.

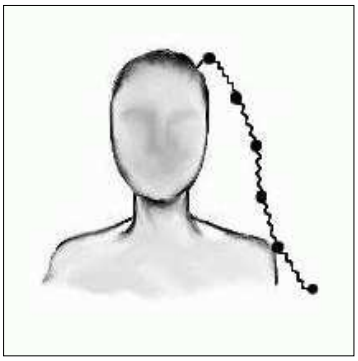


FIG. 1.4 – Collision entre une chaîne masses-ressorts et le corps

Animation d'un milieu continu

Hadap et Magnenat-Thalmann [6] ont d'abord proposé de modéliser les cheveux comme les lignes de courant d'un champ fluide. La dynamique des fluides est appliquée sur un flux orienté de haut en bas, tandis que le corps du personnage est modélisé comme un obstacle pour le flux. Mais avec ce type de méthode, les cheveux gardent toujours la même orientation (ils ne se croisent jamais), et il est donc impossible de simuler des mouvements faisant intervenir des collisions entre cheveux.
Cette méthode a été améliorée en proposant [7] d'utiliser pour la chevelure un modèle d'animation double : un premier modèle, à grande échelle, traduit les diverses interactions en présence au cours du mouvement d'une chevelure dans l'air en considérant cette chevelure comme un milieu continu, soumis aux lois de la dynamique des fluides ; à petite échelle, on modélise la dynamique propre d'un cheveu individuel en tenant compte de sa géométrie et de sa raideur. Le mouvement complet de la chevelure se compose donc d'un mouvement global, régi par la dynamique des fluides, et de mouvements locaux de quelques cheveux individuels visant à accroître le réalisme visuel de l'animation.

1.1.3 Techniques de gestion de la complexité

Le problème majeur de l'animation de chevelures vient du fait que, d'une part, le nombre de cheveux constituant une chevelure est très important (environ 100 000 cheveux composent une chevelure réelle), et d'autre part, que l'ensemble des interactions en présence est très complexe ; en effet, la chevelure interagit avec le corps, et les cheveux interagissent également entre eux. Etant donné le nombre total de cheveux composant une chevelure, on comprend bien qu'il est totalement impossible, en des temps de calcul raisonnables, d'animer individuellement chaque cheveu, et en même temps de détecter et de traiter toutes les collisions qui se produisent au cours du mouvement.
Comment peut-on alors réduire la complexité du système à animer ? Si certaines méthodes développées consistent à animer les cheveux de manière individuelle, tout en utilisant des algorithmes très simples et efficaces de gestion des collisions, d'autres génèrent le mouvement de la chevelure à partir de l'animation d'un nombre restreint d'éléments, et privilégient ainsi des algorithmes gérant les interactions

de manière plus évoluée.

Gestion du nombre d'éléments à animer

Malgré la puissance accrue des ordinateurs, on ne peut pas espérer pouvoir animer individuellement tous les cheveux composant une chevelure en des temps de calcul raisonnables, même en négligeant toutes les interactions.
Les méthodes basées sur l'animation individuelle de cheveux [2,7,9,13] se sont donc toujours limitées à l'animation de quelques dizaines de milliers de cheveux. Anjyo *et al.* [9] ont animé 20 000 cheveux sans traiter les collisions entre cheveux, à raison de 50 secondes par image (cf. fig. 1.5). En utilisant du calcul parallèle, Hadap et Magnenat-Thalmann [7] ont réussi à animer 10 000 cheveux en prenant en compte les intercollisions, à raison de 2 minutes par image.
Les méthodes alternatives à celles qui animent les cheveux de manière individuelle s'appuient sur le fait que la chevelure présente une cohésion, globale ou locale. Certains travaux [2] ont proposé d'interpoler la chevelure autour de quelques cheveux directeurs, d'autres [11, 12, 16] de diviser la chevelure en un nombre limité de mèches que l'on anime séparément.
Dans le paragraphe suivant, on examine comment, à partir de ces deux types de méthodes, les interactions entre la chevelure et le corps, et au sein même de la chevelure, ont pu être modélisées.

Gestion des interactions

Afin de simplifier la détection des collisions entre la chevelure et le corps, une méthode couramment employée [9,15] consiste à approcher le corps par un ensemble de formes géométriques simples, comme des ellipsoïdes. Afin d'obtenir des résultats plus précis et plus naturels, d'autres méthodes considèrent le modèle geométrique du corps sans l'approcher, et, pour éviter d'avoir à calculer les collisions entre la chevelure et des milliers de polygones, utilisent des algorithmes d'accélération de calcul : Lee et Koo [13] utilisent une grille précalculée qui stocke, pour chaque direction (ϕ, θ) de l'espace la distance r de la surface extérieure du corps à partir de l'origine. Le test de collision revient alors à une simple comparaison de distance. Plante *et al.* [16] (qui modélisent la chevelure sous forme d'un ensemble de mèches) se servent d'une grille de l'espace englobant le corps et la chevelure, et qui stocke, dans chacune de ses cellules, la liste des polygones ainsi que la liste des segments de mèches intersectant la cellule considérée, et ce à chaque instant. Le test de collisions consiste donc simplement à tester l'interpénétration de chaque segment de mèche avec les polygones du corps situés dans la même cellule de la grille (*i.e.* dans la même région de l'espace).
Les interactions entre cheveux sont, de manière générale, beaucoup plus complexes à gérer : il est d'une part difficile de connaître exactement la nature des forces que les cheveux exercent réellement les uns sur les autres ; d'autre part, si l'on anime les cheveux de manière individuelle, la gestion de ces interactions suppose de tester un nombre considérable d'éléments. C'est pour cette raison que, dans les méthodes où les cheveux sont animés individuellement, les interactions au sein de la chevelure sont modélisées de manière très simplificatrice.
Les paragraphes qui suivent présentent les différentes méthodes utilisées pour gérer les interactions entre cheveux.

Non prise en compte des interactions entre cheveux

La méthode développée en 1992 par Anjyo *et al.* [9], qui propose d'animer les cheveux de manière individuelle, néglige les interactions entre cheveux, pour des raisons de coût en temps de calcul ; seules

les collisions entre la chevelure et la tête sont prises en compte. Les chevelures animées de cette façon apparaissent alors très lisses, sans discontinuité aucune (cf. fig. 1.5).

FIG. 1.5 – Animation d'une chevelure négligeant les interactions entre cheveux [9]

Modélisation du volume de la chevelure

Afin de tenir compte de ces interactions, tout en évitant de procéder à la détection des collisions entre cheveux, étape très lourde en temps de calcul, certains chercheurs ont considéré que l'effet essentiel de ces interactions était de donner du *volume* à la chevelure. Ainsi, Kim et Neumann [10] ont modélisé les interactions entre cheveux en donnant artificiellement du volume à la chevelure et n'ont animé réellement que les cheveux situés à la surface de la chevelure. Lee et Koo [13] ont quant à eux étendu et amélioré la méthode [9], en prenant en compte les interactions entre les cheveux de manière efficace : l'algorithme consiste, pour chaque cheveu, à modéliser l'ensemble des cheveux situés en-dessous de lui par une surface dont la distance à la surface du corps représente l'épaisseur de la chevelure à cet endroit ; les collisions sont alors traitées de la même manière que les collisions entre les cheveux et le corps.
Ces méthodes parviennent bien à synthétiser des coiffures statiques réalistes, mais le fait de considérer que l'interaction entre cheveux a pour seul effet de donner du volume à la chevelure est très limitant lorsque l'on se place dans le cas d'une chevelure en mouvement. Notamment, avec une telle approche, il est impossible de restituer les discontinuités locales qui se produisent au sein d'une chevelure en fort mouvement (cf. fig. 1.1), et qui sont pourtant essentielles dans l'apport de réalisme.

Interpolation de la chevelure autour de quelques cheveux directeurs

Si l'on considère que la chevelure présente une cohésion *globale* dans son aspect pendant le mouvement, on peut alors se contenter d'animer uniquement quelques cheveux directeurs, puis d'étendre ce mouvement au reste de la chevelure par interpolation. Cette technique a été proposé il y a une dizaine d'années [2]. L'inconvénient est qu'elle ne s'applique qu'à des chevelures très lisses, et qui, encore une fois, ne présentent aucune discontinuité lors du mouvement. En outre, cette méthode pose le problème des collisions entre la chevelure et le corps, puisque seuls les quelques cheveux directeurs sont capables de détecter une interpénétration.

Cheveux plongés dans un milieu continu

Hadap et Magnenat-Thalmann [7] gèrent la complexité des interactions entre cheveux, entre les cheveux et l'air, et entre les cheveux et le corps en traitant l'ensemble de la chevelure comme un milieu continu, régi par les lois de la dynamique des fluides. Le résultat est assez convaincant concernant le volume des chevelures ainsi animées. Cependant, comme le montre la figure 1.6, cette méthode conduit encore à des animations de chevelures très lisses, où n'apparaissent pas les discontinuités que l'on observe dans le cas réel (cf. fig. 1.1).

FIG. 1.6 – Animation de la chevelure comme un milieu continu [7]

Regroupement des cheveux en mèches

Une autre idée, est de considérer que la chevelure présente des cohésions internes *locales*. Ainsi, la chevelure est modélisée comme un ensemble de *mèches*, que l'on anime séparément. Cela revient en fait à se placer à un niveau de détail plus grossier que le cheveu individuel, et permet donc de réduire fortement le nombre d'éléments individuels à animer, tout en maintenant à la chevelure une certaine complexité.
Plusieurs chercheurs [10–12] ont exploité un tel modèle, mais seuls Plante *et al.* [16] ont utilisé des modèles physiques de mèches volumiques déformables pour l'animation. Leur méthode propose une modélisation prenant en compte l'essentiel des phénomènes d'interaction, et ce, pour des chevelures de longueur quelconque. Les résultats obtenus, bien que lourds en temps de calcul, sont visuellement réalistes (cf. fig. 1.7). Dans la suite, cette méthode nous servira de *méthode de référence*

Nous allons donc décrire ici ce modèle, puisque c'est lui que nous allons principalement exploiter dans la suite (cf. 2).
Dans ce modèle, chaque mèche est structurée en une hiérarchie de représentations simples :

- un squelette extensible, qui définit le mouvement de la mèche à grande échelle ;
- une enveloppe volumique déformable, modélisant l'épaisseur de la mèche autour du squelette ;
- des cheveux individuels, qui sont dessinés sur la mèche au moment du rendu, en fonction des déformations de l'enveloppe.

L'interaction entre cheveux proches est prise en compte dans la modélisation de la chevelure par mèches *déformables* : en effet, les forces de frottement auxquelles sont soumis les cheveux appartenant à une même mèche influent sur la forme de la mèche au cours du mouvement.

Grâce à la modélisation de la chevelure sous la forme d'un ensemble de mèches, on peut désormais gérer les collisions internes à la chevelure non pas à l'échelle du cheveu, mais à l'échelle de la mèche : on étudie les collisions non pas entre cheveux, mais *entre mèches*.
Afin de garantir la continuité du mouvement de la chevelure, tout en permettant d'obtenir des discontinuités locales (cf. fig 1.1), le modèle proposé pour les interactions entre mèches, appelé *modèle d'interactions anisotropes*, traite les collisions de deux manières différentes, en fonction de l'orientation des mèches en jeu :

- si les mèches sont quasi-parallèles, on considère qu'il n'y a pas véritablement collision, mais plutôt une continuité dans le mouvement. Les mèches s'interpénètrent, et sont soumises à des forces de frottement visqueux ;
- si les mèches ont des orientations différentes, on considère qu'il y a collision, avec discontinuité dans le mouvement. Les deux mèches se repoussent, de manière très dissipative (pas de rebond).

L'inconvénient de cette méthode par rapport aux objectifs de notre projet, est qu'elle n'est pas du tout interactive : les résultats d'animation présentés fig. 1.7 ont en effet nécessité un temps de calcul d'environ 40 secondes par image (soit 1000 fois plus que le temps de génération d'une image en temps réel[3]). Précisons que plus de 90% du temps a été consacré au traitement de collisions, et environ 68% de ce pourcentage à la détection de collisions entre mèches.

[3]Les images d'un fi lm vid´eo sont transmises `a la fr´equence standard de 25 Herz, soit toutes les 40 millisecondes.

FIG. 1.7 – Résultats de simulation obtenus par [16] pour des chevelures de longueur variable

1.2 Animation par modèles physiques

Pour ce projet d'animation de chevelures, nous avons choisi d'opter pour une animation à base de *modèles physiques*. En effet, comme nous allons le voir dans la suite, l'avantage de ces modèles est qu'ils sont générateurs du mouvement. Dans notre cas, où l'on souhaite réaliser des simulations de mouvement portant sur des chevelures les plus variées possible, ces modèles nous semblent donc bien appropriés.
Nous allons voir ici plus en détail en quoi consistent les modèles physiques.

1.2.1 Approche par modèles physiques

Définition

L'approche par *modèles physiques* d'un phénomène dynamique consiste à étudier ce phénomène afin d'en extraire les *causes* qui génèrent le mouvement. Les modèles physiques permettent de donner des lois régissant une dynamique, indépendamment de la forme de l'objet qui est en mouvement : on ne décrit pas le mouvement d'un objet en particulier ; au contraire, on génère un type de mouvement en modélisant les causes à l'origine de ce mouvement. Le mouvement généré [14] pourrait idéalement s'appliquer à des formes quelconques. Les modèles physiques font ainsi partie d'une famille plus générale de modèles appelés *modèles générateurs* du mouvement.
En ce sens, cette approche se différencie nettement des approches *descriptives*, où *phénoménologiques*, qui consistent à analyser un phénomène en particulier, et à en décrire le comportement. Par exemple, de tels modèles peuvent fournir une fonction d'évolution de l'objet dans le temps, avec un certain nombre de contraintes portant sur les variables dont l'évolution temporelle est corrélée (autrement dit, on fournit une loi *cinématique* de l'objet en mouvement). Le temps apparaît de manière explicite dans de tels modèles, alors qu'il est implicite dans une approche par modèles physique (autrement dit, on fournit une loi *dynamique du mouvement*).
Prenons l'exemple d'une balle rebondissant sur le sol. Une approche descriptive consisterait à analyser le phénomène se produisant pour une balle donnée, dans des conditions particulières. On fournirait ensuite une fonction d'évolution de la position de la balle dans le temps (autrement dit, sa trajectoire), ainsi qu'une fonction décrivant la déformation de la balle au cours du temps. Ce modèle serait bien *complet* (il prendrait en compte l'essentiel des phénomènes observés), et permettrait de restituer le phénomène observé. Mais en aucun cas on ne serait capable, avec un tel modèle, de générer le mouvement de chute d'un autre objet, de forme complètement différente par exemple.
Etudions maintenant, pour le même exemple, une approche par modèles physiques. Dans ce cas, on n'étudie pas une expérience particulière de chute avec rebond, mais *le phénomène en général* de chute d'objet déformable. Il s'agit d'extraire de ces observations les causes qui sont à l'origine des effets que l'on observe. Cette analyse nous permet alors de concevoir un modèle générateur de ce type de mouvement. Ici, on peut par exemple modéliser la balle par un système déformable, et le sol par un système rigide. La balle a une masse, et elle est soumise à des forces physiques de la part de l'extérieur : la force d'attraction gravitationnelle, et la force de contact au sol. Ce modèle n'est pas à coup sûr complet (*i.e.* il ne permet peut-être pas d'obtenir tous les effets voulus), mais il a l'avantage d'être simple et surtout générique du type de mouvement observé.
Au vu de cet exemple, les modèles physiques apparaissent alors comme des modèles qui se situent *en amont* des modèles descriptifs. Ils sont d'autant plus puissants et utiles que le phénomène à modéliser est complexe.

Entités et interactions

Un système dynamique donné est toujours modélisé [14] grâce à deux types d'éléments indissociables :

– les comportements propres de base ;
– les interactions.

Les *comportements propres de base* font référence aux comportements intrinsèques des entités minimales qui composent le système. Par entité minimale, on entend particule élémentaire issue d'une chaîne de décompositions récursives du système en sous-systèmes. L'entité minimale est la limite de cette décomposition, elle n'est elle-même pas décomposable. Deux cas se présentent alors pour l'entité minimale : il peut s'agir soit d'une *masse ponctuelle*, soit d'un *solide élémentaire indéformable*.
Le comportement de base d'une entité minimale est donné dans le cas général par une loi ne faisant intervenir que l'entité considérée. Par exemple, le principe fondamental de la dynamique $m\frac{\partial^2 x}{\partial t^2} = \sum f_i$ est une loi de comportement propre pour l'entité x.
Les *interactions* entre entités minimales sont décrites par les opérateurs *forces*, et par le principe d'*action-réaction*. Les interactions élémentaires peuvent être de deux natures différentes : les interactions *élastiques* (ou *potentielles*) s'expriment en fonction de la position des entités en interaction, tandis que les interactions *visqueuses* font intervenir leurs vitesses.

1.2.2 Intégration du mouvement

Equations en jeu

Nous avons vu que l'approche par modèles physiques d'un phénomène faisait intervenir des équations du mouvement issues de la physique. Or le comportement d'un système dynamique peut toujours [14] être formalisé à l'aide d'équations différentielles linéaires du second ordre, du type :

$$M\ddot{x} + Z\dot{x} + Kx = 0$$

En faisant intervenir des variables supplémentaires, on peut se ramener à un système de deux équations différentielles linéaires du premier ordre couplées. Pour simplifier, nous nous intéressons donc à la résolution d'une équation différentielle linéaire du premier ordre, sous la forme :

$$\dot{x} = f(x,t) \tag{1.1}$$

où f est une fonction connue (*i.e.* qu'on peut évaluer à partir de x et de t), x est l'état du système au temps t, et $\dot{x}$ sa dérivée temporelle. De plus, on suppose connu l'état initial x_0 du système, au temps t_0.

Remarquons que, dans ce projet, les lois physiques qui nous intéressent sont celles gouvernant la dynamique de particules newtonniennes. Dans ce cas particulier, la loi de comportement propre d'une particule est donnée par la seconde loi de Newton :

$$m\ddot{\mathbf{x}} = \mathbf{f}$$

Pour convertir cette équation du second ordre en deux équations du premier ordre couplées, on fait intervenir la vitesse $\mathbf{v}$ de la particule considérée. Cela nous donne le système d'équations :

$$\begin{cases} \dot{\mathbf{v}} = \mathbf{f}/m \\ \dot{\mathbf{x}} = \mathbf{v} \end{cases} \tag{1.2}$$

La position $\mathbf{x}$ et la vitesse $\mathbf{v}$ peuvent être concaténées [1] pour former un vecteur à 6 dimensions. L'espace produit de l'espace des positions et de celui des vitesses (de dimension 6) est appelé *espace des phases*. La résolution du système 1.2 se fait alors dans cet espace.

Des méthodes standard de résolution d'équations différentielles s'intéressent aux solutions symboliques des équations du type 1.1. Par exemple, l'équation différentielle $\dot{x} = -kx$ admet comme solution $x = Ae^{-}kt$.
A l'opposé, puisque notre but ici est de réaliser des simulations par ordinateur, nous allons nous intéresser aux solutions *numériques* de ces équations. Cela signifie que le temps est discrétisé en *pas de temps*, et qu'à chaque pas de temps, on évalue l'état x du système, au moyen d'un *schéma numérique*.

La résolution discrète de l'équation 1.1 (qui fait intervenir des variables dépendant continûment du temps) pose le problème de l'*approximation* de solution. En effet, on souhaite avoir une approximation de la solution analytique valable pour un pas de temps Δt le plus grand possible (afin de limiter le nombre de calculs), mais le risque de *divergence* de la solution numérique par rapport à la solution analytique est à prendre en compte.
Dans les deux paragraphes suivants, nous présentons les principaux schémas numériques existants, et nous examinons les critères sur lesquels il faut se baser pour choisir un schéma numérique qui convient au cas traité (se reporter à [1, 4, 17] pour plus de détails).

Méthodes explicites

Les méthodes numériques dites *explicites* ou *progressives* consistent à calculer l'état du système au temps $t + \Delta t$, à partir de l'état au temps t. Ils se basent sur une approximation du *développement en série de Taylor*, donné par :

$$x(t + \Delta t) = x(t) + \Delta t \frac{\partial x}{\partial t}(t) + \frac{\Delta t^2}{2!}\frac{\partial^2 x}{\partial t^2}(t) + \frac{\Delta t^3}{3!}\frac{\partial^3 x}{\partial t^3}(t) + ... + \frac{\Delta t^n}{n!}\frac{\partial^n x}{\partial t^n}(t) + ... \qquad (1.3)$$

Le *schéma d'Euler* est le schéma numérique le plus simple. Il consiste à calculer l'état x du système au temps $t + \Delta t$ par la formule suivante :

$$x(t + \Delta t) = x(t) + \Delta t f(t) \qquad (1.4)$$

On remarque que 1.4 n'est autre que l'approximation de 1.3 au premier ordre. Ce schéma est donc très simple, mais peu précis : l'erreur d'approximation est en $O(\Delta t^2)$. En particulier, si le pas de temps n'est pas pris suffisamment petit, il y a risque de divergence de la solution. Cependant, un avantage non négligeable du schéma d'Euler est qu'il est économique en temps de calcul.
La méthode du *point milieu* permet de gagner en précision, l'erreur d'approximation qu'elle génère est en $O(\Delta t^3)$. Le schéma consiste à :

1. calculer le pas d'Euler $\Delta x = \Delta t f(x, t)$, grâce à 1.1 ;
2. évaluer f au point milieu par $f_{mid} = f(\frac{x+\Delta x}{2}, \frac{t+\Delta t}{2})$;
3. faire un pas en utilisant la valeur au point milieu : $x(t + \Delta t) = x(t) + \Delta t f_{mid}$.

Ce schéma (également appelé *schéma de Runge-Kutta d'ordre 2*) est donc plus précis que celui d'Euler, mais il nécessite le calcul de deux évaluations de f, au lieu d'une.
En évaluant f un plus grand nombre de fois, on parvient à éliminer les termes des dérivées d'ordre supérieur, et donc à obtenir des schémas encore plus précis. Le schéma le plus connu basé sur ce principe est le *schéma de Runge-Kutta d'ordre 4*. Son erreur d'approximation est en $O(\Delta t^5)$, et il nécessite le calcul de quatre évaluations de f.

Méthodes implicites

Le fait de remplacer le problème exact par un problème discrétisé (donc approché) implique en général des perturbations sur les données ; si on n'est pas assuré que de "petites" perturbations sur les données entraînent des perturbations "pas trop grandes" sur la solution, il devient très difficile de calculer une solution approchée du problème. On parle alors de problème *mal conditionné*, c'est-à-dire [4] que les méthodes numériques explicites ne parviennent pas à fournir pour un coût raisonnable une approximation satisfaisante de sa solution.
Dans le cas d'un problème mal conditionné (on parle aussi de problème *raide*), on a alors recours aux méthodes dites *implicites* ou *rétrogrades*. Ces méthodes se basent également sur une approximation de 1.3, mais elles calculent l'état du système au temps t grâce à des évaluations de f en l'état du système au même temps t.
Le schéma implicite le plus simple est le *schéma d'Euler implicite*, qui s'écrit :

$$x(t + \Delta t) = x(t) + \Delta t f(t + \Delta t) \tag{1.5}$$

Cette équation prend tout son sens si on se représente que l'évolution du système est réversible dans le temps : cela signifie alors que si l'on se trouve dans l'état $x(t + \Delta t)$, et que l'on fait le pas $-\Delta t f(t + \Delta t)$ (on recule dans le temps), on se retrouve dans l'état $x(t)$. C'est pour cette raison que ces schémas sont dits *rétrogrades*.
De même que précédemment, les schémas implicites existent aux ordres supérieurs. Ces schémas, qui conduisent à la résolution de systèmes matriciels (ce qui est plus coûteux qu'un calcul explicite), fournissent dans la majorité des cas une solution d'approximation acceptable (avec le l'amortissement numérique plutôt qu'une tendance à la divergence), et ce, en utilisant des pas de temps beaucoup plus grands que ceux utilisés pour les schémas explicites.

1.2.3 Gestion des collisions

Les problèmes liés aux collisions et au contact interviennent très fréquemment lors d'une simulation un peu complexe, et sont très difficiles à résoudre. Comme nous l'avons vu en 1.2.1, les entités minimales d'un système sont modélisées, soit par des masses ponctuelles, soit par des solides élémentaires indéformables. Il suffit donc d'étudier les collisions se produisant entre de tels éléments.
Le problème de collision peut être décomposé en deux sous-problème : la *détection* de collision et la *réponse* à la collision.

Détection

Etant donné que la simulation progresse à pas discrets, une collision est détectée lorsque l'on détecte une interpénétration de deux éléments. La méthode la plus rigoureuse consisterait alors à calculer l'instant où il y a effectivement eu collision, puis de ramener tout le système à cet instant. En pratique, bien que cela soit moins précis, il s'avère plus facile de déplacer simplement l'élément en

interpénétration jusqu'à la surface de collision.
Notons que si le système est composé d'un grand nombre d'entités minimales, alors il peut s'avérer préférable, au lieu de tester toutes les entités, d'utiliser une *hiérarchie de volumes englobants* autour des différentes parties du système pour détecter plus rapidement s'il y a collision.

Réponse

Afin de décrire la réponse à une collision, on décompose les vitesses et les vecteurs forces en leurs composantes normales et tangentielles par rapport à la surface de collision.
Le type de collision le plus simple à considérer est la collision élastique sans frottement, puisque, comme nous l'avons vu en 1.2.1, celle-ci ne fait intervenir que les positions des éléments en collision. La force d'interaction élastique entre deux particules élémentaires, de positions respectives x_1 et x_2 est de la forme :

$$F_{1->2} = k_e(x_1 - x_2)$$

où k_e est une constante appelée *raideur*.
Lorsque la collision est de type frottement visqueux, la vitesse des éléments en collision intervient proportionnellement à une constante de viscosité. La force d'interaction visqueuse entre deux particules élémentaires, de vitesses respectives v_1 et v_2 est de la forme :

$$F_{1->2} = k_v(v_1 - v_2)$$

où k_v est une constante appelée *constante de viscosité*.

Une méthode simple et particulièrement adaptée à la collision entre objets rigides (qui se traduit par un *choc*) est d'agir directement sur la position et la vitesse des objets. Les deux objets détectés en interpénétration sont déplacés de sorte à ne plus s'interpénétrer, et leurs vitesses sont également modifiées de manière à simuler un choc (élastique ou visqueux, selon les cas). Cette méthode revient en fait à appliquer aux objets en collision des forces d'interaction infiniment grandes durant un temps infiniment court.

Chapitre 2

Méthodologie proposée

Cette partie a pour but de développer une méthodologie afin de répondre aux objectifs fixés en introduction.

2.1 Choix d'un modèle

2.1.1 Rappels des objectifs attendus

On souhaite réaliser une animation de chevelure la plus générique possible, la plus réaliste possible, et interactive.

2.1.2 Avantages et limitations de la méthode de référence

Nous avons vu que la méthode de référence a l'avantage de s'appliquer à des types de chevelures différents, et de produire des résultats très réalistes grâce à la prise en compte dans le modèle de l'ensemble des interactions.
Le problème majeur de cette méthode est qu'elle est très coûteuse en temps de calcul. En particulier, on peut remarquer que la modélisation choisie dans cette méthode entraîne les deux problèmes suivants :

- la détection et le traitement des collisions entre les mèches sont très coûteux du fait que l'enveloppe de chacune de ces mèches est déformable au cours du temps ;
- afin de garantir une animation réaliste, il faut que le nombre de mèches utilisées soit important aux endroits où le mouvement de la chevelure est le plus complexe. Or dans cette méthode, le nombre total de mèches composant la chevelure est constant ; pour obtenir l'échelle de réalisme voulue, il faut donc imposer au départ un nombre total de mèches suffisamment important pour les cas où la chevelure subit de forts mouvements, mais générant alors des calculs inutiles lorsque le mouvement de la chevelure est au repos.

Afin de se rapprocher le plus possible du temps réel, tout en préservant un certain degré de réalisme, nous avons ainsi choisi de modifier ces deux paramètres, en construisant un modèle de chevelure tel que :

- le volume de chaque segment de mèche est indéformable, et de forme simple (cylindrique) : ceci dans le but de faciliter la détection des collisions ;
- la répartition des mèches composant la chevelure varie dans le temps, et s'adapte à la complexité du mouvement présent, grâce à un mécanisme double de subdivision/fusion des mèches : une mèche pourra se subdiviser en plusieurs sous-mèches plus fines, et inversement, plusieurs

mèches pourront fusionner pour donner une seule mèche plus épaisse. Ainsi, les calculs concernant la dynamique du mouvement, la détection et le traitement des collisions, se concentreront principalement sur les zones en mouvement, où la chevelure sera finement subdivisée, tandis que peu de calculs seront demandés par les parties au repos.

Certes, le fait d'imposer un volume cylindrique constant à chaque segment de mèche peut sembler très simpliste, et faire craindre une perte importante du réalisme de l'animation. Cependant, étant donné que, dans notre modèle, les mèches sont autorisées à se subdiviser en sous-mèches plus petites, ou à se fusionner en mèches plus grosses, les déformations locales et les changements de géométrie de la chevelure seront bien prises en compte, venant ainsi compenser la simplification du modèle de chaque mèche.

Notons par ailleurs que, dans ce projet, l'animation et le rendu de la chevelure constituent deux étapes bien distinctes, et que l'essentiel de notre travail porte sur l'*animation* de la chevelure. L'étape de rendu, dont l'étude précise sort du cadre de ce projet, est néanmoins fondamentale pour représenter graphiquement la chevelure de manière la plus réaliste possible. A terme, un habillage géométrique adapté des mèches cylindriques utilisées pour l'animation pourra être utilisé pour accroître le réalisme visuel de l'animation.

2.2 Représentation de la chevelure

2.2.1 Modèle d'une mèche

Similairement à la méthode de référence, nous représentons une chevelure sous la forme d'un ensemble de mèches, chaque mèche étant constituée de trois couches (cf. fig. 2.1) :

- un *squelette extensible*, modélisé par une chaîne de masses ponctuelles (appelées aussi particules) reliées par des ressorts amortis. Un morceau élémentaire du squelette, comprenant deux masses ponctuelles liées par un ressort amorti, sera appelé *segment* de mèche. Les deux masses ponctuelles appartenant à un même segment de mèche sont donc soumises à une interaction de nature élastique amortie ;
- une *enveloppe cylindrique* autour de chaque segment de mèche, et extensible uniquement dans la direction longitudinale (de manière à toujours envelopper chaque segment de mèche). Le rayon de cette enveloppe modélise l'épaisseur du segment de mèche, et va permettre d'utiliser directement le squelette pour la détection de collisions (cf. 2.4.1), et donc de rendre beaucoup plus rapide cette détection ;
- un habillage géométrique et/ou alternatif de l'enveloppe, dont l'étude sort du cadre de ce stage, mais qui fera néanmoins l'objet d'une réflexion en conclusion.

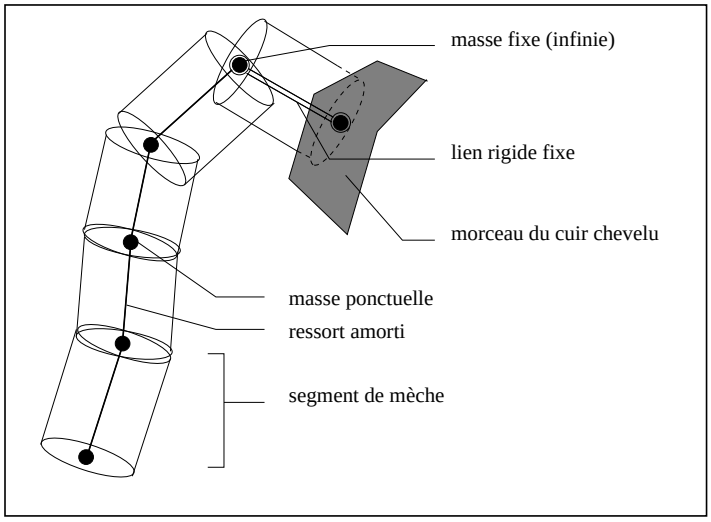


FIG. 2.1 – Modélisation d'une mèche

De même que pour la méthode de référence, le premier segment de la mèche, qui pénètre à l'intérieur de la tête du personnage, est rigide et fixe par rapport à la tête : il ne sera pas pris en compte dans la simulation, mais servira à maintenir l'origine du premier segment simulé hors de la tête. De plus, ce segment rigide permet de transmettre à la mèche le mouvement de la tête.

2.2.2 Structure de la chevelure

Dans notre modèle, la répartition des mèches composant la chevelure est variable : chaque mèche peut être subdivisée en plusieurs sous-mèches, et inversement, deux mèches peuvent être fusionnées en une seule mèche (cf. 2.5). La structure du squelette de la chevelure peut alors être représentée sous la forme d'un *graphe*, chaque nœud du graphe modélisant une masse ponctuelle du squelette, et chaque arête représentant la liaison élastique (modélisée par un ressort) entre deux masses ponctuelles. Afin de définir un sens pour le parcours des arêtes (intuitivement, des racines vers les pointes), l'utilisation d'un graphe *orienté* semble adaptée à la représentation de la chevelure.

FIG. 2.2 – Image réelle d'une chevelure libre

Si l'on observe une chevelure réelle libre (ie. dans laquelle les cheveux sont détachés), on peut remarquer que les mèches de cheveux sont d'autant plus fines et nombreuses qu'elles se trouvent bas dans la chevelure (cf. fig. 2.2). La séparation de la chevelure en mèches apparentes se fait donc naturellement selon une *structure hiérarchique arborescente* ; il semblerait alors assez judicieux de se limiter, comme le montre la figure 2.3, à une représentation de la chevelure sous la forme d'un *arbre orienté*[1].

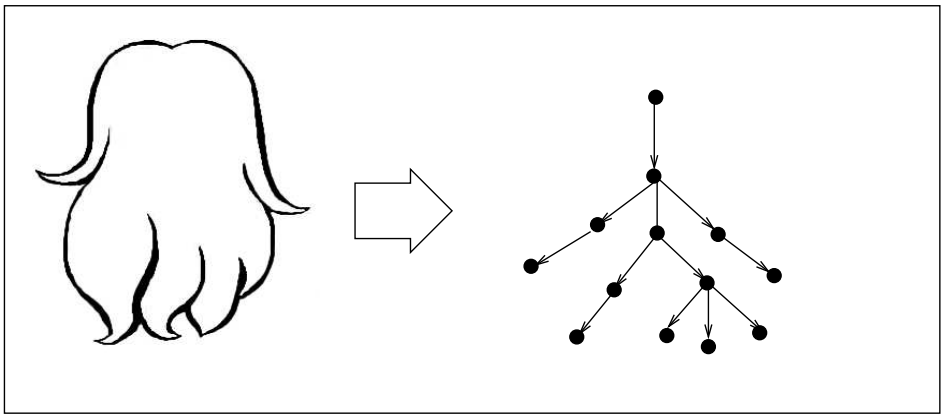


FIG. 2.3 – Structure arborescente d'une chevelure libre

Cependant, si l'on désire pouvoir représenter une chevelure dans le cas le plus général possible, en autorisant ainsi tous les styles de coiffure (queue de cheval par exemple), le modèle plus général de graphe connexe orienté semble mieux convenir. Ce modèle autorise en effet les regroupements de mèches en des points quelconques de la chevelure (cf. fig. 2.4).
En outre, l'algorithme le plus général de fusion des mèches, proposé en 2.5.3, est est stable dans l'espace des graphes connexes, mais pas dans l'espace des arbres, puisque justement il autorise deux

[1]Un arbre orient´e est un graphe orient´e connexe et acyclique.

mèches à se regrouper en un même point pouvant être situé n'importe où au sein de la chevelure.

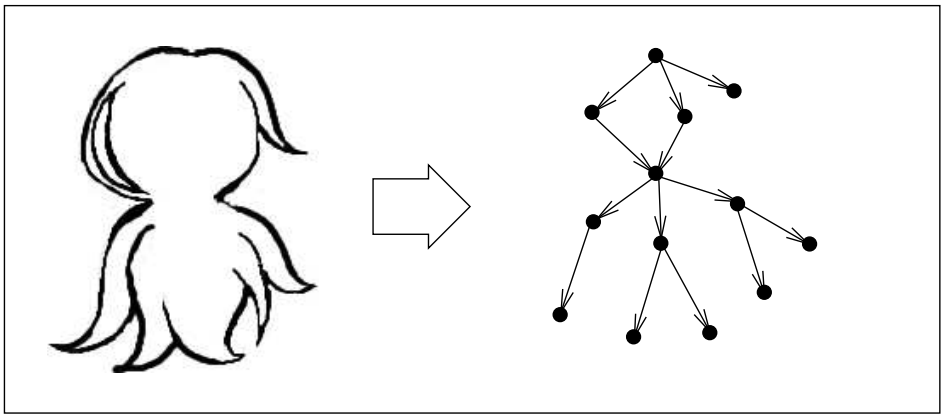


FIG. 2.4 – Représentation d'une chevelure attachée sous la forme d'un graphe connexe orienté

Nous nous limiterons, dans ce rapport, à une description des algorithmes de subdivision et de fusion s'appliquant à des chevelures arborescentes ; les modèles plus généraux, en partie déjà implémentés, finiront d'être développés et testés lors de travaux futurs.

2.3 Gestion des interactions entre les mèches et le corps

2.3.1 Détection de collision

Afin de faciliter la détection des collisions entre les mèches et le corps, le corps du personnage est modélisé uniquement à l'aide de sphères et de cylindres (cf. fig. 2.5).

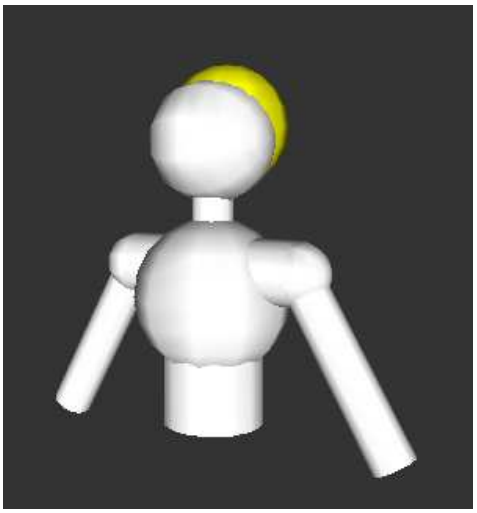


FIG. 2.5 – Modélisation du corps du personnage par un assemblage de sphères et de cylindres

L'ensemble des membres constituant le corps est rangé dans une liste, et pour chaque segment de mèche, on parcourt cette liste et on teste la collision. Cet algorithme a donc une complexité en $O(n.k)$, où n est le nombre de segments de mèches de la chevelure, et k est le nombre de membres constituant le corps. Précisons que si n est variable dans le temps (à cause du mécanisme de division/fusion des mèches), k est fixe et petit ($k = 9$ dans notre exemple).

La manière la plus correcte de procéder pour l'interaction entre le corps et la chevelure serait de tester, pour chaque couple (membre du corps, segment de mèche), s'il y a collision entre le membre (sphère ou cylindre) et le cylindre englobant le segment ; auquel cas il faudrait appliquer un couple de forces aux deux masses situées à chacune des extrémités du segment, en fonction de leur distance au point d'impact. Seulement, si l'algorithme précédent de détection de collision est très simple (les collisions se font entre solides de formes géométriques élémentaires), celui de réponse à la collision est beaucoup plus complexe.
Ainsi, dans un premier temps, nous avons choisi d'utiliser une méthode de gestion des interactions plus simple, consistant à choisir comme enveloppe de chaque segment de mèche la sphère centrée sur l'extrémité destination de l'arête correspondante, et de rayon égal au rayon de l'enveloppe cylindrique théorique.

Cette méthode, quoique inexacte puisque autorisant le type d'interpénétration indiqué sur la figure 2.6, s'avère cependant satisfaisante dans le cas particulier de la collision entre les mèches et le corps (cf. 3.4.1). Ceci est dû au fait que les membres du corps sont globalement nettement plus épais que les segments des mèches, et donc lorsque l'enveloppe cylindrique du segment interpénétre le corps, c'est également le cas pour l'une des deux sphères centrées aux extrémités du segment.

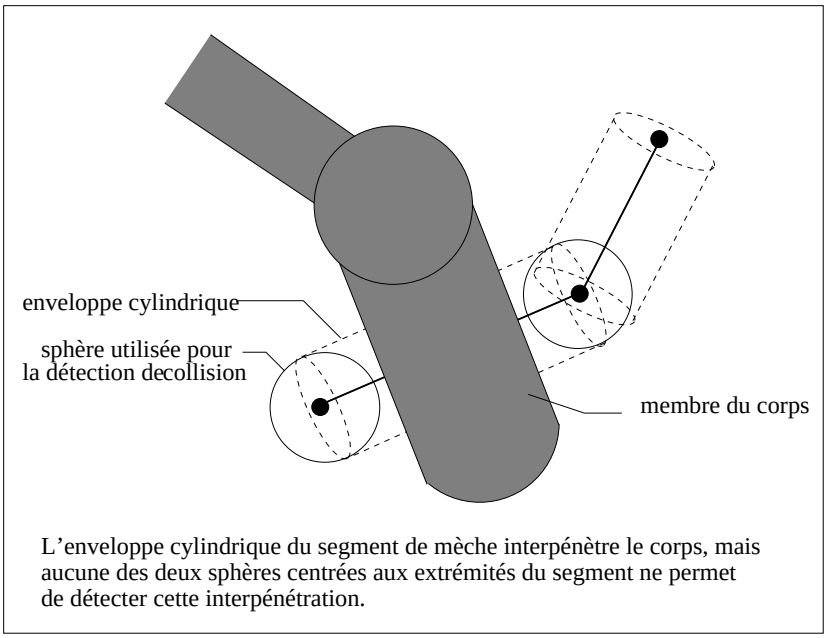


FIG. 2.6 – Cas d'interpénétration non traitée entre le corps et un segment de mèche

2.3.2 Réponse aux collisions

En interaction avec les cheveux, très légers, le corps se comporte comme un objet rigide, et l'on peut négliger les forces exercées par les cheveux sur le corps. De plus, les effets de cette interaction sur les cheveux sont de nature très dissipative (les cheveux ne rebondissent pas sur le corps, mais au contraire s'écrasent dessus).
Il convient alors de modéliser l'interaction entre le corps et la chevelure par des chocs inélastiques dissipatifs. en modifiant directement la position et la vitesse de la mèche détectée en interpénétration. La mèche détectée en interpénétration est alors ramenée hors du corps, par modification directe de sa position, et sa vitesse est également modifiée de manière à simuler un rebond très dissipatif.
Cette méthode est illustrée fig. 2.7 et fig. 2.8, selon que la mèche collisionne un membre du corps cylindrique ou sphérique.

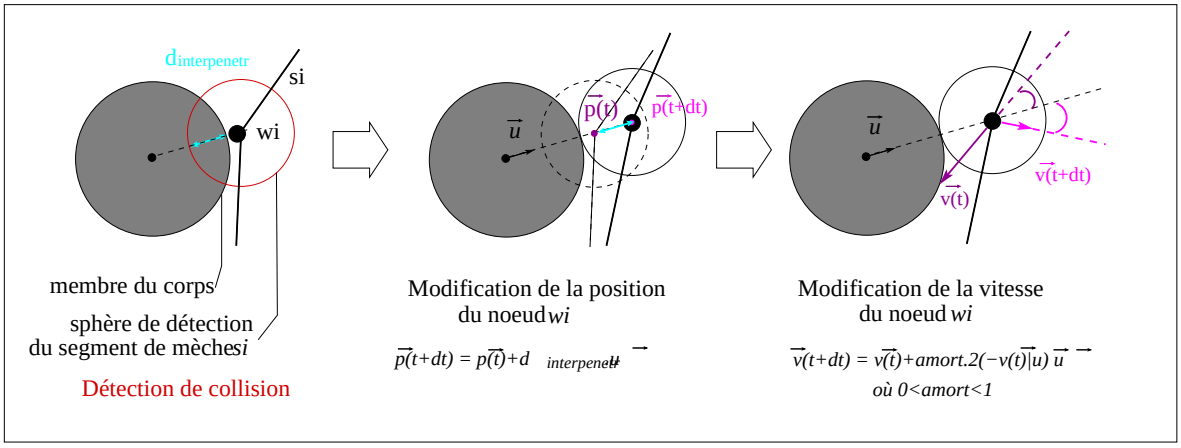


FIG. 2.7 – Réponse à la détection de collision entre une mèche et un membre sphérique

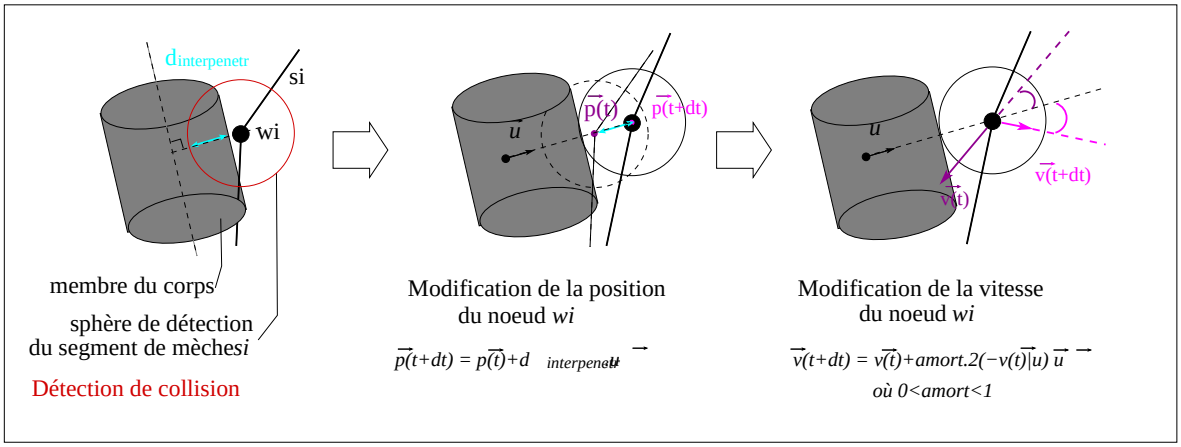


FIG. 2.8 – Réponse à la détection de collision entre une mèche et un membre cylindrique

2.4 Gestion des interactions entre mèches

2.4.1 Détection de collision

L'algorithme naïf de détection des interactions entre mèches consiste à parcourir l'ensemble des segments de la chevelure, et pour chacun de ces segments, tester s'il y a collision avec une autre arête du graphe. Sa complexité est donc en $O(n^2)$, où n est le nombre total de segments de la chevelure.
Si la chevelure comporte un grand nombre de segments ($n \geq 50$) algorithme est donc bien trop coûteux en temps de calcul pour permettre une animation interactive, et il convient de développer un algorithme capable d'accelérer cette détection. Pour ce faire, nous utilisons, comme dans la méthode de référence, une grille régulière de l'espace, qui stocke dans chaque région 3D discrétisée l'ensemble des segments qui intersectent cette région. Cette méthode est détaillée dans la suite.
Quant au test de détection en lui-même, il est rendu beaucoup plus direct et rapide par notre méthode (en comparaison avec la méthode de référence), puisque l'enveloppe de nos segments est modélisée par un cylindre indéformable. Le test de détection est illustré fig. 2.9.

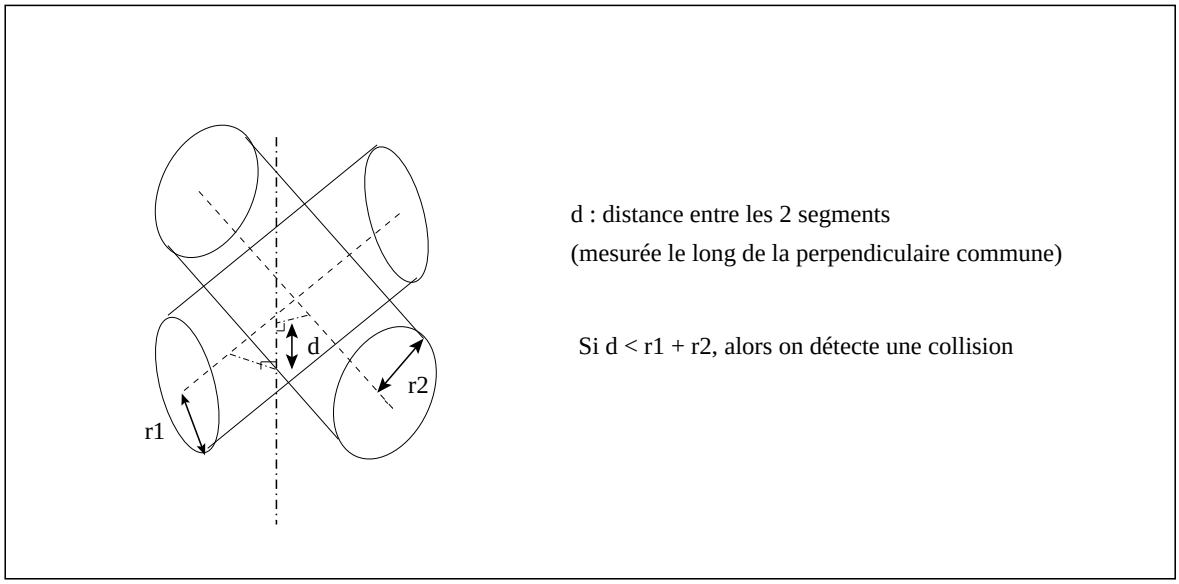


FIG. 2.9 – Détection de collision utilisant les enveloppes cylindriques

Optimisation du parcours des segments susceptibles de s'intersecter

Comme nous l'avons précisé plus haut, un parcours naïf des segments de la chevelure conduirait à un coût quadratique.
Pour réduire ce coût, il paraît judicieux d'utiliser une grille 3D mise à jour à chaque pas de temps, et dont chaque cellule contient à chaque instant la liste des segments qui intersectent la portion de l'espace correspondante. A chaque pas de temps, on teste alors uniquement l'interpénétration des segments se trouvant dans la même cellule de la grille, ce qui limite considérablement le nombre de tests à effectuer, en particulier lorsque la chevelure comporte un grand nombre de mèches.

La grille 3D est représentée par une table de hachage, associée à la fonction de hachage[2] suivante :

$$index : pos3D \longrightarrow ptr\_cell(pos3D)$$

où $pos3D$ est un vecteur position de l'espace, et $ptr\_cell(pos3D)$ l'indice de la cellule de la grille 3D contenant cette position.

L'algorithme utilisant la grille 3D est le suivant : A chaque pas de temps, on effectue les opérations suivantes :

1. initialisation de la grille : on parcourt le graphe de la chevelure, et on rentre chaque segment dans la grille 3D (i.e. pour chaque segment, les cases situées à proximité de chacune des extrémités sont marquées comme contenant ce segment ;
2. détection de collision : pour chaque arête e_i :
   - on construit la liste des cellules de la grille susceptibles d'être intersectées par e_i. Concrètement cette liste est formée par l'union des deux blocs de cellules situées autour des deux extrémités de e_i, chaque bloc ayant une amplitude en nombre de cellules proportionnelle à l'épaisseur du cylindre englobant l'arête e_i. Ces blocs de cellule sont obtenus grâce à la fonction de hachage appliquée aux extrémités de l'arête ainsi qu'aux positions voisines dans l'espace.
   - on parcourt ensuite cette liste de cellules, et pour chaque cellule on parcourt les sommets w_j stockés. on teste alors, pour chacune de ces arêtes e_j, s'il y a collision entre w_j et e_i.

Notons que par manque de temps, on utilise ici une méthode de détection et de réponse semblable à celle décrite en 2.3.1 (détection sphère/sphère au lieu de cylindre/cylindre). Si cette méthode pouvait se justifier dans le cas de la collision avec le corps, étant donné que les membres du corps sont en général beaucoup plus larges que les segments de mèche, ici ce n'est pas le cas. Puisque les mèches peuvent se subdiviser en sous-mèches plus fines, il est très possible d'avoir des segments de mèches très longilignes, et à ce moment-là, il y a risque de ne pas appliquer de forces à des mèches qui s'interpénétrent pourtant.
Dans la prolongation de ces travaux, une méthode plus précise et plus rigoureuse de traitement des collisions sera employée, basée sur la recherche du point le plus proche entre les deux segments de mèche détectées en interpénétration.

2.4.2 Modèle d'interactions anisotropes

Le modèle d'interactions anisotropes défini dans la méthode de référence sera utilisé ici. Ainsi, deux mèches quasi-parallèles s'interpénétreront et seront soumises à des forces de frottement visqueux, tandis que deux mèches non parallèles seront soumises à un choc élastique.
La transition entre les deux modes d'interaction se fait de manière contrôlée, grâce à l'automate d'état suivant :

[2] De mani`ere g´en´erale, une fonction de hachage f associ´ee `a la table de hachage g associe une clef (de type quelconque) `a un indice (entier) de h.

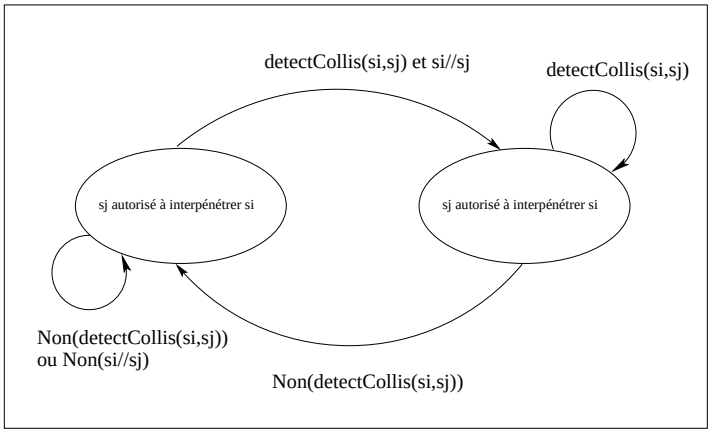


FIG. 2.10 – Automate d'état pour le segment s_j

2.5 Subdivisions et fusions adaptatives

2.5.1 Modèle adaptatif

La méthode de référence fournit, comme nous l'avons vu, des résultats très satisfaisants, mais lourds en temps de calcul (plusieurs heures de calcul pour quelques secondes d'animation). Afin de se rapprocher d'une simulation en temps réel, notre but est ici de voir comment il est possible, à partir d'un tel modèle, de construire un modèle *adaptatif*, c'est-à-dire capable d'adapter automatiquement, par un mécanisme double de subdivision/fusion la répartition de la chevelure en mèches plus ou moins fines, en fonction de la complexité de l'évolution du système à chaque pas de temps. Pour cela, on est amené à définir un certain nombre de *critères* sur lesquels on va se baser pour quantifier cette complexité.
Précisons que les algorithmes de subdivision/fusion présentés ici s'appliquent à une chevelure modélisée par un arbre orienté. L'avantage, pour l'instant, de se limiter à des chevelures arborescentes, est de pouvoir proposer deux mécanismes de subdivision et de fusion à la fois simples et antagonistes. Un algorithme plus général de fusion que nous avons développé (cet algorithme n'est pas stable dans l'espace des arbres) sera évoqué en conclusion.

2.5.2 Subdivision d'une mèche

Mécanisme de subdivision

La subdivision d'une mèche en plusieurs sous-mèches implique :

- une modification locale de la structure de la chevelure ;
- une répartition correcte des différents paramètres dynamiques de la mèche initiale (masse, volume et raideur) entre les nouvelles sous-mèches ;
- un traitement des interactions adapté entre les nouvelles sous-mèches.

FIG. 2.11 – Image réelle d'une chevelure au repos

FIG. 2.12 – Mouvement violent d'une chevelure réelle

FIG. 2.13 – Mouvement rapide d'une chevelure réelle dans la direction transverse à l'orientation des mèches

FIG. 2.14 – Mouvement rapide d'une chevelure réelle dans la direction d'orientation des mèches

Critères de subdivision

Nos critères de subdivision portent sur les nœuds du graphe représentant la chevelure. Afin de choisir des critères de subdivision pertinents, nous nous sommes basés sur l'observation du phénomène réel du mouvement de chevelure.
En comparant une chevelure au repos (cf. fig. 2.11) et une chevelure en fort mouvement (cf. fig. 2.12), on remarque que, la complexité visuelle du mouvement de la chevelure est grande lorsqu'une vitesse importante est communiquée à la chevelure, alors qu'elle est quasi-inexistante lorsque cette chevelure

est au repos.
D'autre part, si l'on examine les figures 2.13 et 2.14, on constate que la séparation de la chevelure en plusieurs sous-mèches est beaucoup plus nette lorsque la chevelure subit un mouvement transverse par rapport à son orientation. On peut voir aussi que la séparation d'une mèche en plusieurs sous-mèches à tendance à s'effectuer du bas de vers le haut de la chevelure.
Toutes ces remarques nous amènent à choisir de subdiviser une mèche de notre modèle en plusieurs sous-mèches lorsque l'on détecte un nœud n vérifiant les critères de subdivision suivants :

- n est une feuille de l'arbre, ou bien n a au moins deux fils. Ceci revient à dire que pour engendrer une subdivision, n ne doit pas avoir doit soit correspondre à une pointe de la chevelure, soit se trouver au niveau d'une "jointure" entre une mèche supérieure et un ensemble de mèches inférieures résultants de la subdivision de la mèche supérieure. Cela permet de garantir que les mèches se sépareront toujours du bas vers le haut de la chevelure.
- la vitesse de n est grande, i.e. de norme supérieure à un *seuil* de subdivision dont la valeur sera fixée lors des tests ;
- la direction de la vitesse de n est à peu près perpendiculaire à l'orientation locale de la mèche (cf. fig.)

Effets de la subdivision

Lorsqu'un nœud n vérifie les critères de subdivision, la structure du graphe est alors modifiée, comme l'illustre la figure ci-dessous :

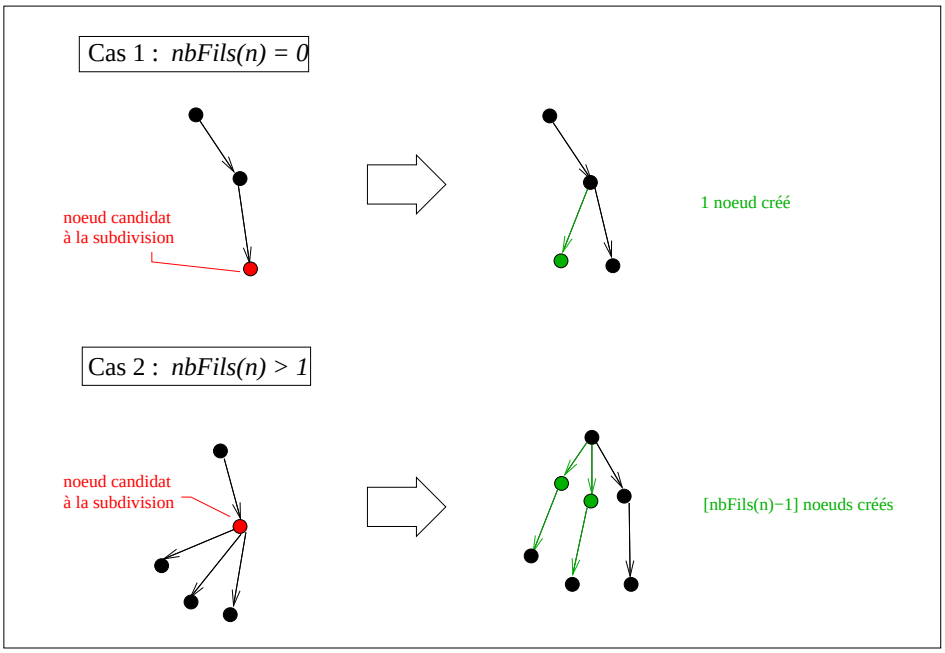


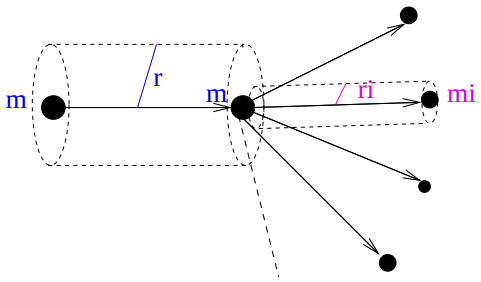
FIG. 2.15 – Modification locale de la structure de l'arbre orienté lors de la subdivision

Précisons qu'il ne se produit jamais de subdivision longitudinale (il n'y a pas d'ajout de nœud le long d'un même segment de mèche), mais qu'au contraire, toutes les subdivisions sont transverses.
Les positions et vitesses des nouveaux nœuds ainsi créés sont fixées de manière à ce que les nouveaux

segments de mèches créés restent dans le plan de la vitesse du nœud n détecté candidat à la subdivision, et s'étalent légèrement dans le sens de la vitesse, comme on peut l'observer sur la figure 2.13. Ensuite, afin de respecter les principes de conservation de masse, et de volume chacun des nouveaux nœuds créés, se voit attribuer une masse et un rayon tels que le graphe respecte toujours les règles suivantes :

$$\text{conservation de la masse : } m = \sum mi$$

$$\text{conservation du volume : } r^2 = \sum ri^2$$



Notons que, après subdivision, les sous-mèches sont pratiquement totalement confondues, et donc l'union de leurs volumes ne correspond pas, à ce moment-là, au volume de la mèche initiale. Afin de "laisser le temps" aux sous-mèches de se séparer physiquement, et d'éviter ainsi une transformation trop brutale du volume, la transmission du volume de la mèche d'origine aux sous-mèches se fait de manière continue ; c'est-à-dire que les volumes des sous-mèches passent progressivement du volume d'origine à la valeur finale de ce volume, par interpolation linéaire sur leur rayon. La durée de cette phase transitoire devrait, idéalement, être réglée de manière à coïncider avec le temps de séparation des mèches. Puisque notre rendu sommaire consiste pour l'instant à visualiser les cylindres des mèches, cette méthode a également l'avantage d'éviter les discontinuités visuelles.

Un autre paramètre important à régler correctement est la *raideur* des ressorts ajoutés dans le graphe. En remarquant que la force $F = -kx$ appliquée sur la masse m doit être égale à la somme des forces $F_i = -k_i x_i$ appliquées aux noeuds fils n_i, on a la relation suivante qui est toujours vraie :

$$\text{cohérence des raideurs : } k = \sum ki$$

Enfin, du point de vue de la dynamique, les nouvelles sous-mèches sont traitées comme des mèches quasi-parallèles qui s'interpénètrent ; elles sont donc soumises à des forces de frottements visqueux.

2.5.3 Fusion des mèches

Notre mécanisme de fusion des mèches se déroule exactement à l'inverse du mécanisme de subdivision : la chevelure se "referme " en mèches fusionnées de haut en bas. La fusion de plusieurs mèches se déclenche lorsque plusieurs arêtes ayant la même origine ont leur sommets destinations proches en terme de position et de vitesse.

Critères de fusion

Nos critères de fusion portent sur les arêtes du graphe représentant la chevelure. Plusieurs arêtes ne peuvent se fusionner que si :

– elles sont issues du même sommet ;
– leur proximité en terme de position et de vitesse est très grande.

Effets de la fusion

La structure locale de la chevelure est modifiée comme l'indique le schéma suivant :

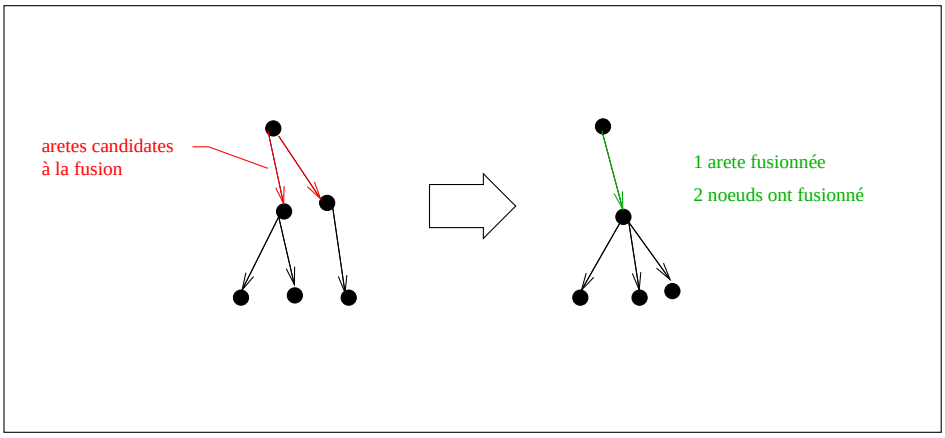


FIG. 2.16 – Modification locale de la structure de l'arbre orienté lors de la fusion

La position (resp. la vitesse) du nœud n résultant de la fusion de tous les nœuds destinations nd_i des arêtes qui fusionnent est calculée comme le barycentre des positions (resp. des vitesses) des nd_i. Les conservations de masse, de volume ainsi que la cohérence du choix des raideurs pour les nouveaux ressorts sont assurées en respectant les mêmes règles que celles énoncées pour la subdivision.

2.5.4 Problème de l'instabilité

Notons enfin que tous les paramètres de seuil devront être réglés avec soin, afin d'éviter qu'une succession non contrôlée de subdivisions et de fusions ne génère des instabilités pour le système.

Chapitre 3

Résultats et bilan

Cette partie présente et analyse les résultats obtenus en utilisant notre méthode, expliquée en 2. Divers tests comparatifs ont été développés afin de valider nos choix de modélisation, et de dresser un bilan sur notre méthode.

3.1 Implantation de la méthode

Pour permettre de valider la méthode décrite en 2, un programme de simulation a été développé en langage C++, en utilisant plusieurs bibliothèques :

- les bibliothèques graphiques `OpenGL` [21] pour l'affichage de la scène, et `Qt` pour l'interface graphique (cf. fig. 3.2) ;
- la bibliothèque `AnimAL` (développée en interne à iMAGIS[1]), qui intègre l'ensemble des fonctionnalités nécessaires à l'animation par modèles physiques (systèmes masses-ressorts, schémas numériques d'intégration du temps, solveurs de systèmes, etc.) ;
- la bibliothèque `QGLViewer` (développée en interne à iMAGIS[2]), basée sur une utilisation de Qt, qui fournit un ensemble d'outils élaborés pour la visualisation de la scène (trackball, sélection et manipulation d'objets, sauvegarde des images, etc.) ;
- la `Standard Template Library`, bibliothèque générique standard du C++ [20], qui fournit un ensemble de structures de données classiques (tableaux, vecteurs, listes etc.) ainsi que les algorithmes associés (parcours, tri, ajout et suppression d'éléments, etc.) ;
- la `Boost Graph Library` [19], qui fournit un ensemble de structures et d'algorithmes permettant de manipuler des graphes.

L'ensemble du développement et des tests s'est effectué sous Linux, sur un PC muni d'un Pentium 4 à 1.7 GHz.

3.1.1 Structure du programme

L'application a été conçue en utilisant une approche orientée objet. Ainsi, le programme est structuré en un ensemble de classes.

[1] La biblioth`eque C++ aniMAL [5] (Animation Algorithms Library) est d´evelopp´ee `a iMAGIS sous la responsabilit´e de Franç ois Faure.

[2] La biblioth`eque C++ QGLViewer [3] a ´et´e d´evelopp´ee `a iMAGIS par Gilles Debunne de janvier `a mai 2002.

Les objets que l'on manipule et qui interagissent sont des instances de ces classes. Notre application utilise en particulier les objets suivants :

- pour l'interface : un objet Qt `mainWidget` de classe *FMain*, qui contient l'objet *m_wViewer* (fenêtre qui sert à visualiser la scène), ainsi que tous les boutons de réglage ;
- pour le modèle du corps : un objet `body` de classe *Body* qui consiste en un ensemble de sphères et de cylindres ;
- pour le modèle de la chevelure : un graphe `graph` de classe *Graph*, modifiable par les méthodes de subdivision et de fusion. Chaque segment du graphe est représenté par un objet *boundingCylinder* de classe *BoundingCylinder*, qui sert à la détection de collisions ainsi qu'au rendu des mèches ;
- pour l'animation : un objet `anim` de classe *Animator*, qui gère tous les paramètres d'animation ;
- pour le rendu : un objet `rend` de classe *Render*, qui gère tous les paramètres de rendu.

3.1.2 Interface de l'application

La fenêtre principale (qui permet de visualiser la scène) dérive de la classe `QGLViewer` [3], qui fournit tout un ensemble de fonctionnalités très pratiques pour la visualisation (trackball, sélection et manipulation des objets etc.). Le reste de l'interface est constitué de boutons de réglage pour divers paramètres d'animation et de rendu.

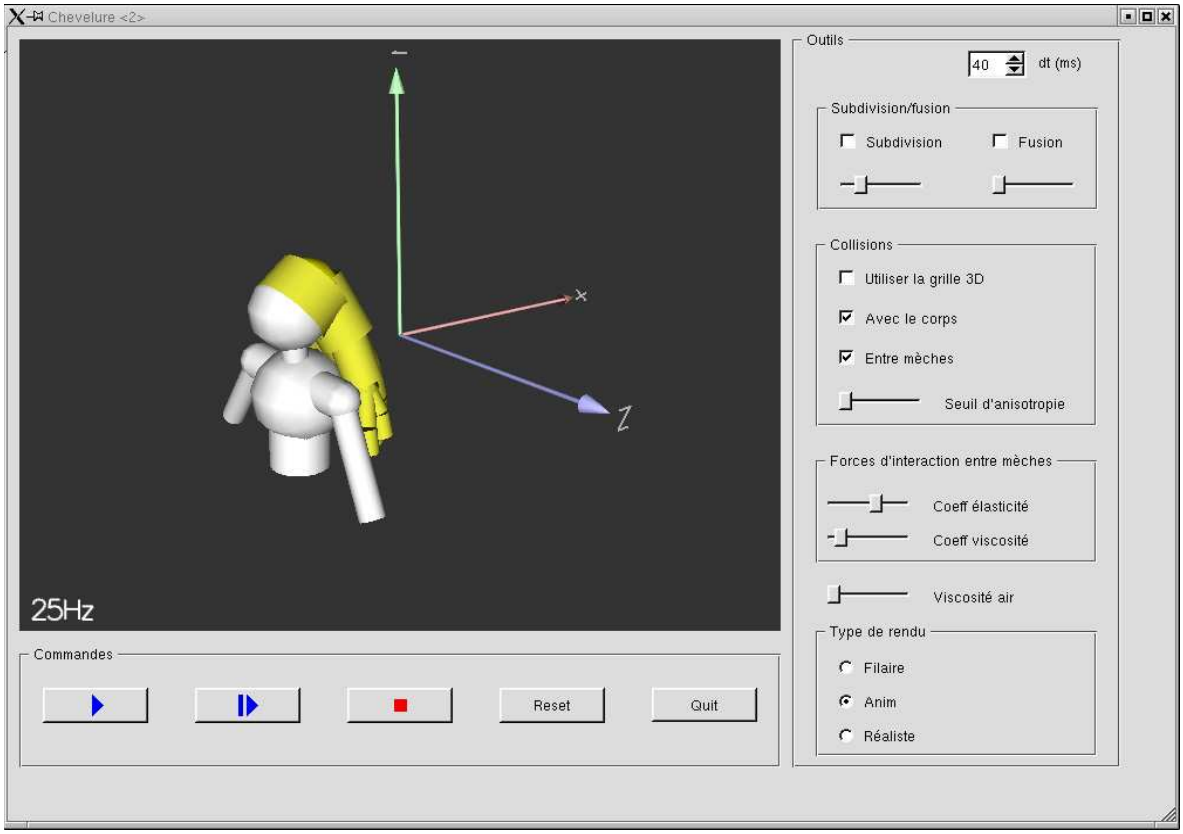


FIG. 3.1 – Interface de l'application

3.2 Méthodologie suivie pour les tests

Nous détaillons ici notre approche pour exploiter, analyser et évaluer notre méthode de la manière la plus pertinente possible. La plus grande difficulté réside dans le fait de trouver un bon compromis entre *temps de calcul* et *réalisme* de l'animation.
Précisons qu'il s'agit ici de réalisme du mouvement, et non pas réalisme du rendu. Par ailleurs, toutes les images de simulation présentées dans ce rapport ont été calculées en utilisant un rendu très sommaire (affichage d'un cylindre par segment de mèche, identique à l'enveloppe cylindrique du segment), afin de simplifier la compréhension de la structure sous-jacente. Notre application permet également de visualiser directement le graphe, ce qui est pratique pour savoir à chaque pas de temps quelle est sa structure.
Notons que l'étape de rendu est très rapide (2 ms par pas de temps), on pourra donc la négliger par rapport au temps de calcul de l'animation.

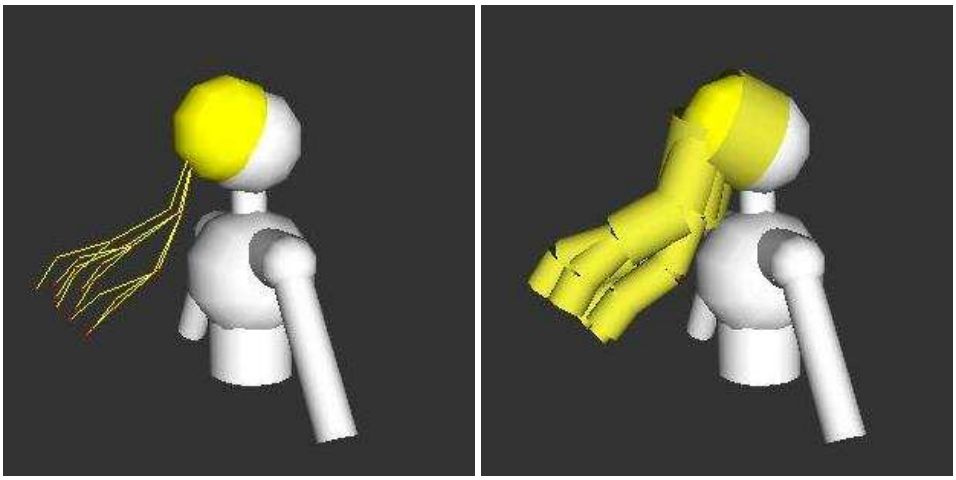


FIG. 3.2 – Affichage du squelette de la chevelure (à gauche), et des enveloppes de segments de mèche (à droite)

3.2.1 Contrainte du temps de calcul

Temps réel et temps interactif

On dit qu'une simulation est en *temps réel* si le temps mis par le processeur pour effectuer un pas de temps est égal au pas de temps fixé pour le modèle d'animation. Cela revient à dire que dans une simulation en temps réel, *le temps dans l'espace de la simulation s'écoule de la même manière que le temps du monde réel* (cf.fig. 3.3).

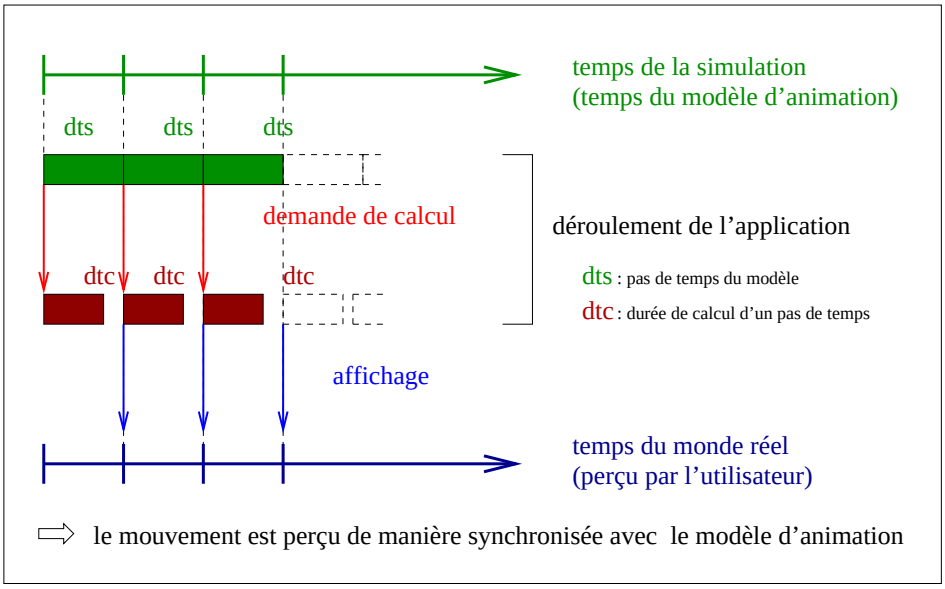


FIG. 3.3 – Déroulement d'une simulation en temps réel

On dira que notre simulation est *interactive* si elle est en temps réel légèrement dégradé. Cela signifie que le temps de calcul d'un pas de temps est légérement supérieur au pas de temps fixé pour l'animation, ce qui entraîne un retard de la perception du mouvement par rapport au modèle (cf. fig. 3.4).

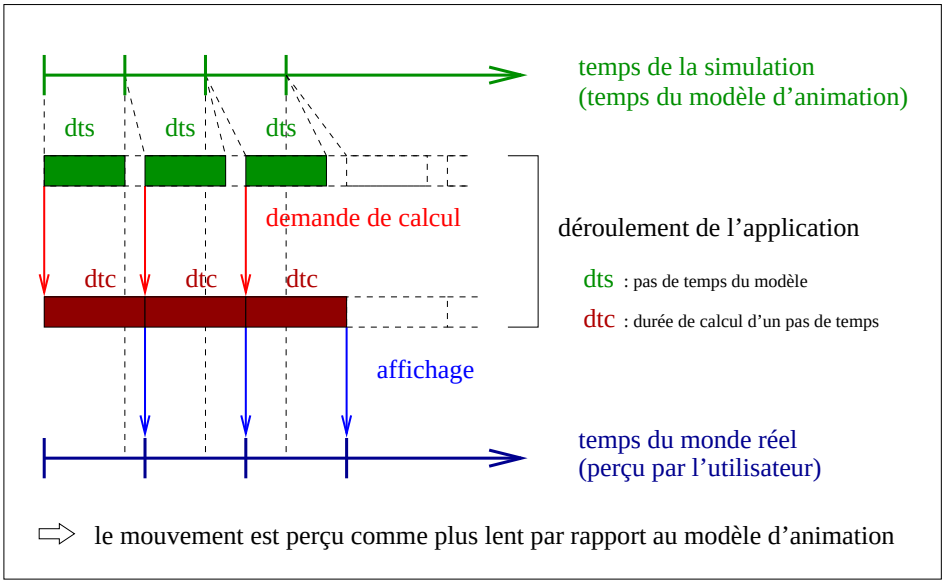


FIG. 3.4 – Déroulement d'une simulation en temps différé

Fréquence d'affichage

Nous choisirons, dans la limite du possible, de générer les images de la simulation à la fréquence de **25 Hz**, qui correspond à la fréquence standard des flux vidéo[3].
Le pas de temps du modèle d'animation sera toujours fixé à **40 ms** ($= 1/25$ ms). Ainsi, si l'on est en temps réel, on aura bien une fréquence d'affichage de 25 Hz. Par-contre, pour une simulation en temps différé, la fréquence sera inférieur à 25 Hz, et le mouvement sera perçu comme saccadé. Les tests porteront sur des simulations strictement temps réel, et également sur des simulations interactives, pour lesquelles on autorisera une fréquence légèrement inférieure à 25 Hz.

Choix d'un schéma d'intégration

Nous testerons les quatre schémas explicites[4] suivants : Euler explicite, Runge-Kutta d'ordre 2, Runge-Kutta d'ordre 4, et le Point Milieu Modifié[5].
Dans notre cas, un bon schéma d'intégration devrait, idéalement, remplir les conditions suivantes :

- être rapide, i.e. générer à chaque pas de temps le moins de calculs possible ;
- supporter un pas de temps grand (en l'occurrence, un pas de temps de 40 ms) ;
- supporter des paramètres physiques proches des paramètres réels.

Le choix d'un bon schéma d'intégration est très délicat, car les trois critères précédents sont fortement liés les uns aux autres, et aucun des schémas numériques que nous avons testés n'est capable d'être le plus performant pour les trois critères à la fois.
Notons que les deux derniers critères sont importants si l'on veut garantir le réalisme physique de la simulation. Le premier critère, quant-à lui, concerne l'efficacité de la simulation, et doit être pris en compte si l'on désire réaliser des animations à la fois complexes (i.e. comportant le plus grand nombre de nœuds possible), et en temps réel.
Le choix du schéma d'intégration va donc se faire en plusieurs étapes. D'ores et déjà, comme il a été précisé dans le paragraphe précédent, nous choisissons de toujours fixer le pas de temps dt du modèle d'animation à **40 ms**. On verra en 3.3 que cette contrainte (qui correspond au deuxième critère), élimine déjà le schéma d'Euler explicite, qui diverge lorsque le pas de temps est aussi grand.

3.2.2 Contrainte du réalisme

Afin que la simulation reflète le plus possible la réalité, divers paramètres du modèle d'animation doivent être réglés correctement :

- les paramètres liés à la dynamique des systèmes masses-ressorts (sans faire encore intervenir les collisions) ;
- les paramètres intervenant dans le réponse aux collisions ;
- les seuils utilisés pour les opérations de subdivision et de fusion des mèches.

Par ailleurs, la complexité du système (i.e. le nombre de nœuds simulés) joue un rôle important dans l'apport de réalisme. Mais bien évidemment, plus le système simulé est complexe, plus la simulation est coûteuse en temps de calcul. Il s'agira alors de déterminer la complexité maximale du système que l'on peut garantir tout en restant proche du temps réel. Les différents schémas numériques, ainsi que

[3] La fr´equence de 25 Hz correspond `a la fr´equence limite pour laquelle l'œil ne perç oit plus le mouvement comme saccad´e, ´etant donn´e que la persistance r´etinienne est d'environ 1/25 ms

[4] Le sch´ema d'Euler implicite, pr´esent´e en 1.2.2, a donn´e des r´esultats m´ediocres, certainement dˆus `a un d´efaut de son implantation ; il n'a donc plus ´et´e test´e par la suite.

[5] Sch´ema explicite d'ordre 4, plus pr´ecis que le sch´ema de Runge-Kutta d'ordre 4. Pour plus de d´etails, se r´ef´erer `a [17].

les algorithmes d'accelération de calcul, seront testés et comparés entre eux, afin d'optimiser cette complexité.

3.2.3 Evaluation du résultat

Notre méthode devra être comparée avec une méthode qui, similairement à la méthode de réference, anime une chevelure composée d'un nombre fixe de mèches ne pouvant ni se subdiviser ni se fusionner (ce modèle correspond en fait à un cas limite de notre propre modèle, dans lequel toutes les mèches sont entièrement subdivisées).
Les critères de comparaison utilisés seront, encore une fois, le temps de calcul et le réalisme de l'animation.

3.3 Dynamique du système

Le tableau comparatif ci-dessous donne, pour chacun des schémas numériques testés, la fréquence d'affichage (en FPS[6]) ainsi que la raideur linéique maximale des ressorts du système qui garantit la convergence du schéma utilisé (au-delà de cette valeur, le schéma diverge).

Schéma numérique	**FPS** (Hz)	**Raideur linéique max** (N/m^2)
Euler explicite	-	-
Runge-Kutta d'ordre 2	286	150
Runge-Kutta d'ordre 4	264	350
Point milieu modifié	260	500

TAB. 3.1 – Performances des différents schémas numériques, pour des simulations effectuées sur un graphe de 28 noeuds (sans la gestion des collisions)

On peut tout d'abord remarquer qu'il est impossible de trouver une raideur linéique acceptable (i.e. supérieure à 1 N/m^2) pour laquelle le schéma d'Euler explicite converge. Cela est dû au fait que le pas de temps choisi pour l'intégration (40 ms) est trop grand pour permettre la convergence de ce schéma, dont l'inconvénient est justement de ne supporter que des pas de temps très petits (cf. 1.2.2). Ce schéma est donc abandonné dans la suite.
Ensuite, si l'on examine l'aspect de la chevelure en fonction de la raideur linéique des ressorts utilisée (cf. fig. 3.5), on constate qu'une raideur linéique de 150 N/m^2 est trop faible pour représenter une chevelure de manière réaliste : avec une telle valeur, les mèches apparaissent comme exagérément extensibles. Par-contre, une raideur linéique supérieure à 350 N/m^2 donne des résultats convenables.

[6]Frame Per Secund, nombre d'images g´en´er´ees par seconde (homog`ene au Hz)

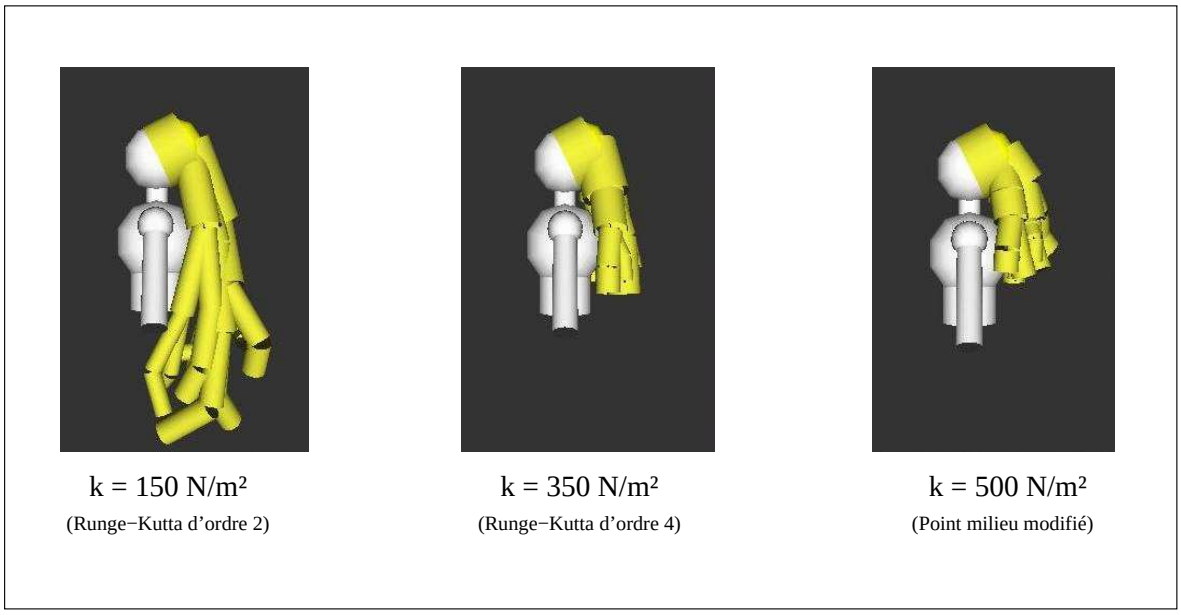


FIG. 3.5 – Aspect de la chevelure en fonction de la raideur linéique maximum supportée par chacun des différents schémas numériques

L'utilisation du schéma de Runge-Kutta d'ordre 2 semble donc difficile, puisqu'il n'est pas possible, avec un tel schéma, d'utiliser des valeurs de raideur linéique correctes sans provoquer la divergence. Nous continuerons tout de même à lui faire subir les tests suivants, à titre comparatif, car ce schéma a quand même l'avantage d'être nettement plus rapide que les deux autres.
Parmi les autres schémas restants, le plus rapide est le schéma de Runge-Kutta d'ordre 4, mais le schéma du point milieu modifié est plus robuste par rapport au choix des paramètres du système. Dans la suite des tests, nous continuerons à utiliser ces deux schémas.

3.4 Gestion des collisions

3.4.1 Collisions entre les mèches et le corps

Dans la plupart des cas, la détection de collisions entre les mèches et le corps se fait correctement. La figure ci-dessous montre quelques exemples de collisions, au repos et en mouvement, pour une chevelure composée de 28 nœuds.

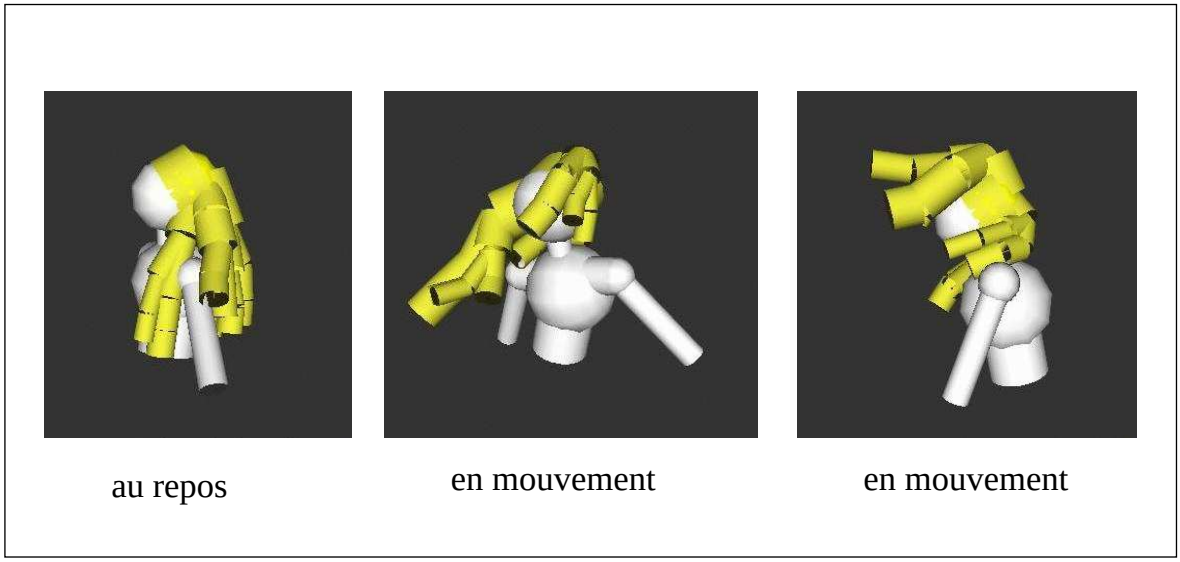


FIG. 3.6 – Exemples de collisions entre la chevelure et le corps

3.4.2 Collisions entre mèches

Réglage des forces

Deux segments de mèches i et j en collision (qui ne sont pas autorisés à s'interpénétrer) sont soumises à des forces de collision élastique F_e de norme $F_e = k_e|d_{ij}|$ où d_{ij} est la profondeur d'interpénétration entre les deux segments. La figure ci-dessous compare les résultats obtenus pour différentes valeurs de la constante k_e.

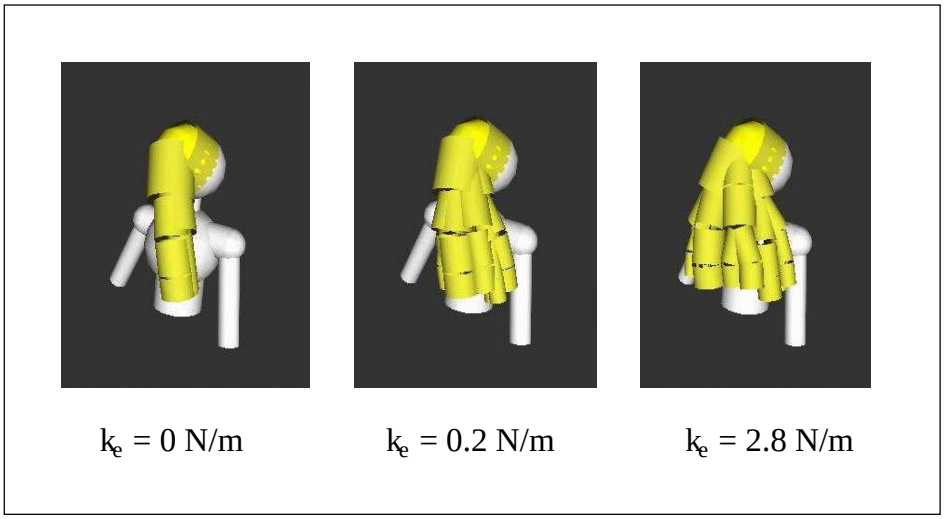


FIG. 3.7 – Aspect de la chevelure en fonction du coefficient k_e de la force de collision élastique

On constate que lorsque k_e est trop faible, les mèches s'interpénétrent, et le volume global de la

chevelure n'est pas conservé. Aussi on choisira toujours dans la suite une valeur de k_e proche de 2 N/m.

3.4.3 Efficacité du calcul

Comparaison des schémas numériques

Schéma numérique	Temps de calcul pour un pas de temps (ms)		
	dynamique du système	collisions avec le corps	collisions entre mèches
Runge-Kutta d'ordre 2	3.4	2.1	18.8
Runge-Kutta d'ordre 4	3.7	2.3	40
Point milieu modifié	3.8	2.6	49.1

TAB. 3.2 – Comparaison du temps de calcul d'un pas de temps en fonction des différents schémas numériques et de la prise en compte des différents types de collisions

Comparaison des algorithmes de détection de collision entre mèches

Dans le paragraphe 2.4.1 de la partie précédente, nous avons mis au point un algorithme de détection des collisions basé sur l'utilisation d'une grille de l'espace. Cet algorithme était censé améliorer le temps de calcul quadratique obtenu avec l'algorithme de parcours naïf.
La figure 3.8, réalisée à l'aide de mesures expérimentales, donne, pour chacun de ces algorithmes, et en utilisant le schéma de Runge-Kutta d'ordre 2, l'évolution du nombre de pas de temps calculés par seconde en fonction du nombre n de nœuds du graphe représentant la chevelure.

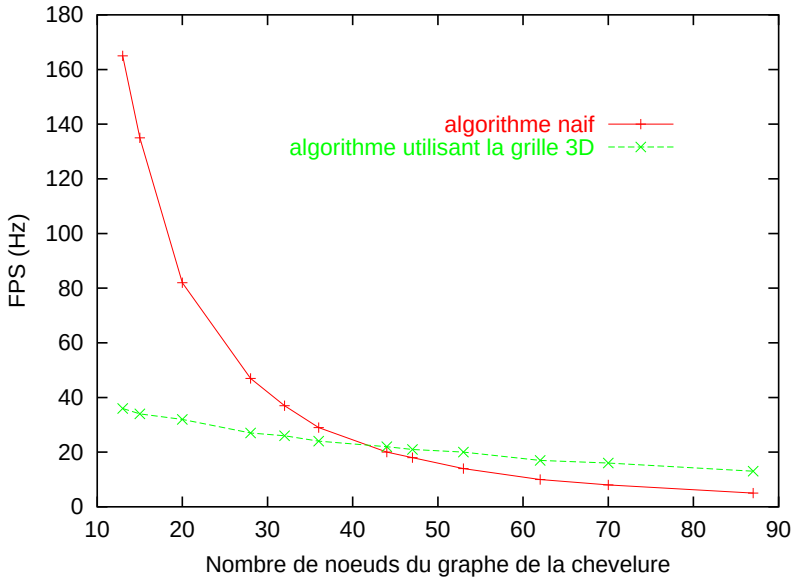


FIG. 3.8 – Comparaison des deux algorithmes utilisés pour la détection de collision entre mèches, en utilisant le schéma d'intégration Runge-Kutta d'ordre 2

Tout d'abord, on peut vérifier que l'algorithme naïf en $O(n^2)$ donne bien une fréquence de calcul en $O(1/n^2)$, alors que pour l'algorithme utilisant la grille 3D, cette fonction est linéaire en fonction de n.
De plus, on constate que les deux courbes se rejoignent à peu près pour un nombre de nœuds égal environ à 35, au niveau de la ligne de fréquence à 25 Hz, correspondant au temps réel. Cela signifie que si l'on désire simuler notre animation strictement en temps réel, d'une part il faut limiter le nombre de nœuds du graphe à 35, d'autre part, les deux algorithmes sont aussi efficaces l'un que l'autre. Par-contre, on s'aperçoit que l'algorithme utilisant la grille 3D permet de simuler un nombre plus important de nœuds tout en restant proche du temps réel, alors que l'algorithme naïf perd toute son efficacité au-delà de 35 nœuds.
Ceci dit, ces tests ont été effectués avec la méthode de Runge-Kutta d'ordre 2, qui est le schéma le plus rapide parmi ceux dont on dispose, et dont on a pu constaté précédemment qu'il ne permettait pas de simuler la chevelure en utilisant des constantes de raideur acceptables. Cela signifie que si l'on privilégie l'un des deux autres schémas, plus robustes, alors l'algorithme utilisant la grille 3D sera moins efficace que l'algorithme naïf, même sur la plage des fréquences interactives. On pourra tout de même utiliser la grille 3D pour accelérer le calcul de simulations mettant en jeu un grand nombre de nœuds, obtenant ainsi des temps de calcul raisonnables et certainement meilleurs que ceux obtenus par la méthode de référence.

3.5 Subdivision et fusion

Dans tous nos tests, les seuils de déclenchement de subdivision et de fusion ont été réglés manuellement. A terme, il faudrait pouvoir fixer précisément les valeurs de ces seuils qui permettent d'obtenir une animation la plus réaliste possible.

fusion

Les images ci-dessous illustrent le mécanisme, très simple, de fusion des mèches.

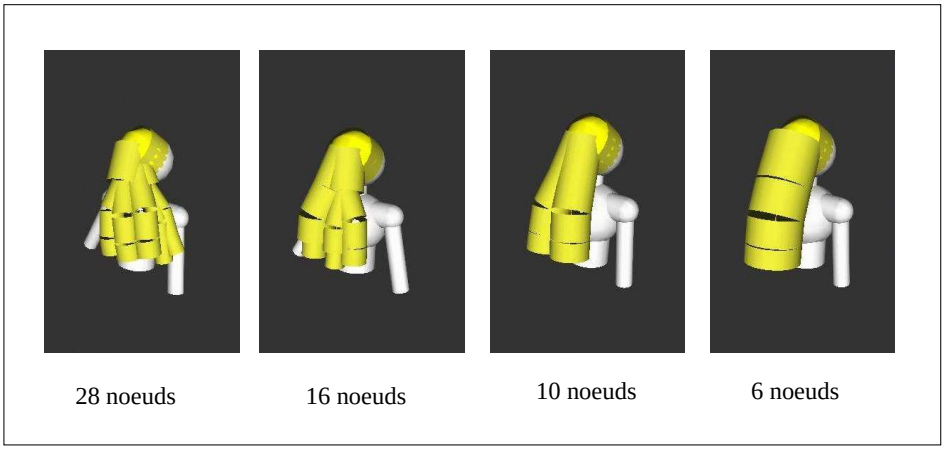


FIG. 3.9 – Fusion des mèches :
la chevelure initiale comporte 28 nœuds, et la chevelure finale 6 nœuds

Subdivision

Le mécanisme de subdivision est illustré par la séquence ci-dessous, qui met également en œuvre le mécanisme de fusion des mèches.

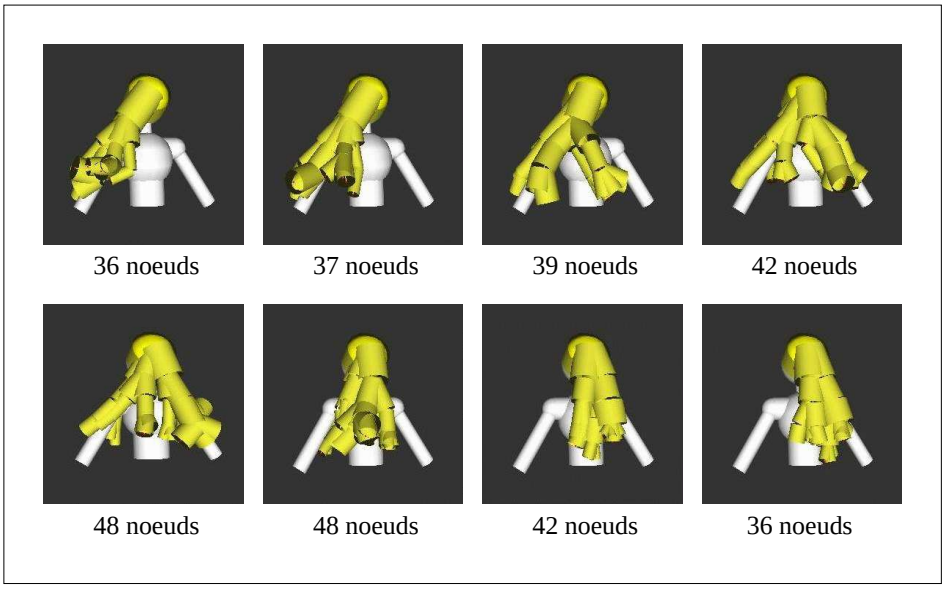


FIG. 3.10 – Subdivision et fusion successives des mèches :
la chevelure comporte 36 nœuds au départ et à la fin

La chevelure réagit bien comme on le souhaitait : il y a multiplication des nœuds du graphe lorsque la chevelure est soumise à un fort mouvement, et fusion des nœuds lorsque la chevelure revient au repos.

Distribution de la complexité

Le principe de notre méthodologie est le suivant :

1. En prenant en compte tous les phénomènes d'animation qui génèrent un temps de calcul non négligeable (i.e. la dynamique et la gestion des collisions), on détermine le nombre optimal nb_{max} de nœuds du graphe tel que la simulation soit proche du temps réel.
2. nb_{max} étant fixé, on répartit ces nb_{max} nœuds au sein de la chevelure grâce au double mécanisme de subdivision/fusion :
   On suppose que le nombre de nœuds du graphe initial est inférieur à nb_{max}. A chaque pas de temps :
   – on réalise la fusion des mèches, et on calcule le nombre n_{fus} de nœuds du graphe (forcément $n_fus \leq nb_{max}$) ;
   – on calcule $n_{disp} = nb_{max} - n_{fus}$, i.e. le nombre de nœuds maximum que l'on pourrait ajouter dans le graphe tout en restant en dessous de la borne nb_{max} ;
   – on lance l'opération de subdivision des mèches en lui passant en argument n_{disp} : la subdivision au pas de temps courant doit générer au maximum n_{disp} nœuds supplémentaires dans l'arbre. Si ce n'est pas le cas, certaines subdivisions qui auraient pu se produire sont

interdites. La meilleure manière de choisir les subdivisions à effectuer parmi l'ensemble des subdivisions candidates consisterait à trouver un algorithme de choix utilisant des critères de subdivision plus précis. En pratique, par manque de temps, notre algorithme est basé sur le principe `FIFO` : les premières subdivisions candidates sont les premières servies.

A terme, le système fonctionnera donc de manière parfaitement autonome : a chaque pas de temps, il sera capable de répartir les nœuds du graphe là où le mouvement est le plus complexe, et simplifier le graphe ailleurs.

Conclusion et perspectives

Lors de ce projet, nous nous sommes intéressés au problème de l'animation à la fois réaliste et interactive de chevelures virtuelles. Aucune méthode, tant dans le monde de l'industrie que dans celui de la recherche, n'est actuellement capable de résoudre de manière satisfaisante ce problème : les animations interactives de chevelure se font au prix d'un réalisme très dégradé, et les méthodes - très récentes - qui permettent de simuler le mouvement réaliste d'une chevelure sont aussi très coûteuses en temps de calcul.

Dans l'optique d'apporter à ce problème une solution meilleure que les méthodes existantes, nous avons mis au point une nouvelle méthodologie d'animation de chevelures par mèches, qui permet d'adapter automatiquement le niveau de résolution des mèches constituant le modèle en fonction de la complexité locale du mouvement que subit la chevelure.
Notre modèle d'animation inclut la gestion de la dynamique des mèches individuelles, la prise en compte des collisions entre les mèches et le corps, et entre mèches, et enfin le double mécanisme de subdivision/fusion des mèches.
Concernant la dynamique du système, nous avons constaté qu'il était difficile de trouver un schéma d'intégration numérique qui soit à la fois très robuste vis-à-vis des paramètres du système (notamment des raideurs des mèches), et rapide. Il pourrait sans doute s'avérer utile, par la suite, de développer un schéma numérique spécialement adapté à notre dynamique.
Comme nous avons pu le remarquer, le temps de calcul consacré à la gestion des collisions est de plus de 80 % par rapport au temps total. L'algorithme de détection de collisions utilisant une grille de l'espace, a montré qu'il n'accelérait pas les calculs pour les zones de fréquence correspondant au temps réel. Cet algorithme pourrait plutôt servir à accelérer le calcul de simulations plus réalistes exécutées en temps différé.
Enfin, les tests effectués sur l'adaptativité du système montrent que la chevelure se comporte bien de la manière souhaitée au cours du mouvement : les mèches qui subissent un fort mouvement se subdivisent en mèches plus fines, tandis que les mèches au repos qui sont proches l'une de l'autre fusionnent pour donner une seule mèche plus grossière. Il reste cependant encore beaucoup de tests à effectuer sur le double mécanisme de subdivision/fusion. En particulier, il faudra essayer de cibler précisément les critères à l'origine d'une subdivision de mèche. De plus, il est souhaitable d'étendre cette méthode à des cas où la chevelure est modélisée de manière plus générale par un graphe orienté, en nous autorisant ainsi à travailler sur des chevelures coiffées par exemple.

Au cours de ce projet, nous n'avons pas développé de modèle de rendu ; ceci est envisagé pour la suite de nos travaux. En effet, il est probable que l'étape de rendu contribue grandement à augmenter la part de réalisme de nos animations. Afin de garantir un coût d'affichage faible, il convient de chercher

des méthodes de rendu économiques, et permettant tout de même de fournir des détails visuels précis. Les méthodes de rendu alternatif semblent bien adaptées au problème. Une première idée serait de représenter chaque segment de mèche à l'aide d'un billboard[7] sur lequel on plaquerait une texture de mèche de cheveux semi-transparente. Afin de garantir la continuité du rendu lorsqu'il se produit une subdivision ou une fusion de mèches au sein de la chevelure, il serait sans doute utile de disposer d'une famille de textures multirésolutions dans laquelle on viendrait puiser en fonction du volume associé au segment de mèche.

[7] Il s'agit d'un support plan et rectangulaire, toujours orient´e normalement `a la vue, et sur lequel on plaque une texture semi-transparente.

Bibliographie

[1] David Baraff and Andrew Witkin. Physically based modeling. In *Siggraph'98 Course Notes*, 1998.

[2] Agnès Daldegan, Nadia Magnenat Thalmann, Tsuneya Kurihara, and Daniel Thalmann. An integrated system for modeling, animating and rendering hair. In R. J. Hubbold and R. Juan, editors, *Eurographics '93*, pages 211–221, Oxford, UK, 1993. Eurographics, Blackwell Publishers.

[3] Gilles Debunne. Qglviewer. http ://www-imagis.imag.fr/Membres/Gilles.Debunne/CODE/QGLVIEWER/, 2002.

[4] M. Crouzeix et A.L. Mignot. *Analyse numérique des équations différentielles*. Masson, 1992.

[5] François Faure. Documentation animal. http ://dev-imagis.imag.fr/AnimAL/, 2002.

[6] Sunil Hadap and Nadia Magnenat-Thalmann. Interactive hair styler based on fluid flow. In *Computer Animation and Simulation'00*, Computer Graphics Forum, pages 87–100. Blackwell Publishing, 2000.

[7] Sunil Hadap and Nadia Magnenat-Thalmann. Modeling dynamic hair as a continuum. In *EG 2001 Proceedings*, volume 20(3) of *Computer Graphics Forum*, pages 329–338. Blackwell Publishing, 2001.

[8] J.T. Kajiya and T.L. Kay. Rendering fur with three dimensional textures. *Computer Graphics*, 23(3) :271–280, july 1989.

[9] Yoshiaki Usami Ken-ichi Anjyo and Tsuneya Kurihara. A simple method for extracting the natural beauty of hair. In *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 111–120. 1992.

[10] Tae-Young Kim and Ulrich Neumann. A thin shell volume for modeling human hair. In *Proceedings of the Conference on Computer Animation (CA-00)*, pages 104–111, Los Alamitos, CA, May 3–5 2000. IEEE Press.

[11] Chuan Koon Koh and Zhiyong Huang. Real-time animation of human hair modeled in strips. In *Computer Animation and Simulation'00*, pages 101–112. 2000.

[12] Chuan Koon Koh and Zhiyong Huang. A simple physics model to animate human hair modeled in 2d strips in real time. In *Computer Animation and Simulation'01*, pages 127–138. 2001.

[13] Doo-Won Lee and Hyeong-Seok Ko. Natural hairstyle modeling and animation. *Graphical models*, 63(2) :67–85, March 2001.

[14] Annie Luciani. Animation par modèles physiques pour la synthèse d'image et la robotique. Cours 2001-2002 du DEA IVR, mars 2002.

[15] Eric Plante. Mouvements et interactions de la chevelure par mèches déformables, décembre 1999.

[16] Eric Plante, Marie-Paule Cani, and Pierre Poulin. A layered wisp model for simulating interactions inside long hair. In Nadia Magnenat-Thalmann Marie-Paule Cani, Daniel Thalmann, editor, *Computer Animation and Simulation 2001 Proceeding*, Computer Science. EUROGRAPHICS, Springer, sep 2001. Proceedings of the EG workshop of Animation and Simulation.

[17] William H. Press, William T. Vetterling, Saul A. Teukolksy, and Brian P.Flannery. *Numerical recipes in C*. Cambridge University Press, 1992.

[18] R. E. Rosenblum, W. E. Carlson, and E. Tripp, III. Simulating the structure and dynamics of human hair : modelling, rendering and animation. *The Journal of Visualization and Computer Animation*, 2(4) :141–148, October–December 1991.

[19] Jeremy Siek, Lie-Quan Lee, and Andrew Lumsdaine. The boost graph library. http ://www.boost.org/libs/graph/doc/index.html, 2000.

[20] Inc. Silicon Graphics Computer Systems. Standard template programmer's guide. http ://www.sgi.com/tech/stl/, 1994.

[21] Mason Woo, Tom Davis, Jackie Neider, and Dave Shreiner. *OpenGL Programming Guide*. Addison Wesley, third edition, 1999.